

**PERAMALAN HARGA CABAI MENGGUNAKAN METODE  
*HIGH ORDER FUZZY TIMES SERIES MULTIFACTORS***

**SKRIPSI**

untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ridho Agung Gumelar

NIM: 135150207111096



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PERAMALAN HARGA CABAI MENGGUNAKAN METODE *HIGH ORDER FUZZY TIMES*  
*SERIES MULTIFACTORS*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh :

Ridho Agung Gumelar

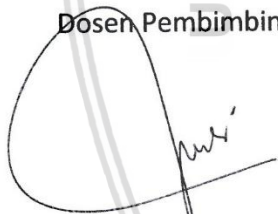
NIM: 135150207111096

Skripsi ini telah diuji dan dinyatakan lulus pada

02 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Budi Darma Setiawan, S.Kom, M.Cs

NIP: 19841015 201404 1 002

Dosen Pembimbing II



Putra Pandu Adikara, S.Kom, M.Kom

NIP: 19850725 200812 1 002

Mengetahui,

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200132 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Agustus 2018



Ridho Agung Gumelar

NIM: 135150207111096

## KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT, karena atas rahmat-Nya penulis dapat menyelesaikan skripsi dengan judul "Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors*".

Dalam menyelesaikan skripsi ini, penulis banyak mendapat bantuan dan masukan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Budi Darma Setiawan, S.Kom., M.Cs., dan Bapak Putra Pandu Adikara, S.Kom., M.Kom., selaku dosen pembimbing yang telah membagikan ilmu serta arahan dalam proses pengerjaan skripsi,
2. Bapak Wayan Firdaus Mahmudy, S. Si., M.T., Ph.D., selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D., selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya,
4. Bapak Agus Wahyu Widodo, S.T., M.Cs., selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya,
5. Bapak Sigit Adinugroho, S.Kom., M.Sc., selaku pembimbing akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi,
6. Kedua orang tua dan keluarga yang senantiasa mendoakan dan memberikan semangat untuk menyelesaikan skripsi,
7. Pihak yang turut membantu untuk menyelesaikan skripsi ini dengan baik.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kesempurnaan, untuk itu kritik dan saran yang bersifat membangun senantiasa penulis harapkan guna kemajuan skripsi kami di kemudian hari.

Malang, 10 Agustus 2018



Penulis

Email: ridhogumelar12@gmail.com

## ABSTRAK

Kebutuhan sehari-hari masyarakat Indonesia tidak lepas dari komoditas pertanian seperti cabai, bawang merah, bawang putih, tomat, dan lain-lain. Beberapa komoditas pertanian tersebut di antaranya memiliki fluktuasi harga yang tajam, seperti cabai. Ketika pasokan cabai di pasar berkurang, harganya dapat jauh melambung tinggi dari harga normal. Sebaliknya ketika pasokan cabai berlebihan, harganya akan turun jauh di bawah harga normal. Hal ini dipengaruhi oleh berbagai faktor seperti musim panen, jumlah produksi, jumlah konsumsi masyarakat, luas lahan panen dan lain-lain. Berdasarkan permasalahan tersebut maka diperlukan suatu metode untuk dapat memperkirakan harga cabai sehingga dapat digunakan untuk pendukung pembuatan keputusan terkait dengan permasalahan harga. Peramalan merupakan salah satu solusi untuk dapat memperkirakan pergerakan harga komoditas cabai. Metode yang digunakan untuk peramalan harga cabai adalah *High Order Fuzzy Times Series Multifactors*. Pada metode ini pembentukan subinterval dilakukan dengan menggunakan *Fuzzy C-means*. Untuk menghitung tingkat kesalahan hasil peramalan pada penelitian ini menggunakan *Mean Square Error* (MSE). Berdasarkan hasil pengujian yang dilakukan, besarnya nilai data *training* dan *order* yang digunakan dalam peramalan tidak menjamin menghasilkan tingkat kesalahan yang rendah. Hasil peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* mendapatkan hasil MSE terbaik sebesar 20374,19.

**Kata kunci:** harga cabai, fuzzy C-means, fuzzy times series, peramalan.



## ABSTRACT

*The daily needs of Indonesian people can not be separated from agricultural commodities such as chili, onion, garlic, tomatoes and others. Some of these agricultural commodities have sharp price fluctuations, such as chili. When the supply of chilli in the market decreases, the price can be soar higher than the normal price. Conversely, when the supply of chili is excessive, the price will be fall well below the normal price. This is influenced by various factors such as the harvest season, the amount of production, the amount of public consumption, the area of the harvest area and others. Based on these problems we need a method to estimate the price of chili so that it can be used to support decision-making related to price issues. Forecasting is one solution to be able to estimate the price movement of chili commodities. The method used to forecast the price of chili is High Order Fuzzy Times Series Multifactors. In this method the formation of subinterval is done by using Fuzzy C-means. For calculate forecasting error results in this reaserch using Mean Square Error (MSE). Based on the results of the test, the value of training data and orders used in forecasting does not guarantee a low error rate. The results of forecasting the price of chili using the method of High Order Fuzzy Times Series Multifactors get the best MSE results of 20374.19.*

**Keywords:** *price of chili, fuzzy C-means, fuzzy times series, forecasting.*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR KODE PROGRAM .....	xv
DAFTAR LAMPIRAN .....	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan.....	3
1.4 Manfaat .....	3
1.5 Batasan Masalah .....	4
1.6 Sistematika Pembahasan .....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 Objek Penelitian .....	10
2.2.1 Cabai.....	10
2.3 Peramalan .....	11
2.4 Deret Waktu ( <i>Times Series</i> ).....	12
2.5 Logika <i>Fuzzy</i> .....	13
2.5.1 Himpunan <i>Fuzzy</i> .....	14
2.6 <i>Fuzzy C-Means</i> .....	15
2.7 <i>Fuzzy Times Series</i> .....	17
2.8 Peramalan dengan <i>High Order Fuzzy Times Series Multifactors</i> .....	18

2.9 Perhitungan Keakuratan .....	23
BAB 3 METODOLOGI PENELITIAN .....	24
3.1 Tipe Penelitian.....	24
3.2 Strategi dan Rancangan Penelitian .....	24
3.2.1 Gambaran Umum Sistem .....	24
3.2.2 Lokasi Penelitian.....	26
3.2.3 Pengumpulan Data.....	26
3.2.4 Peralatan Pendukung .....	27
3.2.4.1 Perangkat Keras .....	27
3.2.4.1 Perangkat Lunak.....	28
3.4 Jadwal Penelitian.....	28
BAB 4 PERANCANGAN.....	29
4.1 Perancangan Proses .....	29
4.1.1 Proses Perhitungan <i>Universe of Discourse</i> .....	30
4.1.1.1 Proses Perhitungan <i>Mean</i> .....	31
4.1.1.2 Proses Perhitungan Simpangan baku .....	32
4.1.1.3 Proses Perhitungan Nilai <i>Dmin</i> dan <i>Dmax</i> .....	33
4.1.2 Proses Perhitungan Jumlah <i>Cluster</i> .....	35
4.1.3 Proses Perhitungan Pusat <i>Cluster</i> .....	36
4.1.4 Proses Pembentukan Subinterval dan <i>Fuzzy Set</i> .....	41
4.1.5 Proses Fuzzifikasi.....	43
4.1.6 Proses Pembentukan <i>Fuzzy Logical Relationship</i> .....	45
4.1.7 Proses Defuzzifikasi .....	47
4.2 Perhitungan Manual.....	50
4.2.1 Perhitungan <i>Universe of Discourse</i> .....	50
4.2.2 Perhitungan Jumlah <i>Cluster</i> (k) .....	52
4.2.3 Perhitungan Pusat <i>Cluster</i> dan Subinterval .....	54
4.2.3.1 Penentuan Pusat <i>Cluster</i> .....	54
4.2.3.2 Perhitungan Batas Subinterval.....	59
4.2.4 Pembentukan <i>Fuzzy Set</i> .....	60
4.2.5 Fuzzifikasi .....	61



4.2.6 Pembentukan <i>Fuzzy Logical Relationship</i> (FLR) .....	63
4.2.7 Defuzzifikasi.....	63
4.3.1 Halaman Input.....	66
4.3.2 Halaman Tampil Data .....	68
4.3.3 Halaman Proses.....	69
4.3.4 Halaman Hasil.....	72
4.4 Perancangan Pengujian.....	75
4.4.1 Perancangan Pengujian Pengaruh <i>Order</i> Terhadap MSE.....	75
4.4.2 Perancangan Pengujian Pengaruh Data <i>Training</i> Terhadap MSE .....	75
4.4.4 Perancangan Validasi Sistem .....	76
BAB 5 IMPLEMENTASI .....	77
5.1 Spesifikasi Sistem .....	77
5.1.1 Spesifikasi Perangkat Keras.....	77
5.1.2 Spesifikasi Perangkat Lunak .....	77
5.2 Implementasi Algoritme.....	78
5.2.1 Implementasi Proses <i>Universe of Discourse</i> .....	78
5.2.1.1 Implementasi Proses <i>Mean</i> .....	78
5.2.1.2 Implementasi Proses Simpangan Baku .....	79
5.2.1.3 Implementasi Proses <i>Dmin</i> dan <i>Dmax</i> .....	79
5.2.2 Implementasi Proses Jumlah <i>Cluster</i> .....	80
5.2.3 Implementasi Proses Pusat <i>Cluster</i> .....	80
5.2.4 Implementasi Proses Pembentukan Subinterval dan <i>Fuzzy Set</i> .....	83
5.2.5 Implementasi Proses Fuzzifikasi.....	85
5.2.6 Implementasi Pembentukan <i>Fuzzy Logical Relationship</i> .....	86
5.2.7 Implementasi Proses Defuzzifikasi.....	88
5.3 Implementasi Antarmuka.....	91
5.3.1 Implementasi Antarmuka Halaman Input.....	91
5.3.2 Implementasi Antarmuka Halaman Tampil Data.....	91
5.3.3 Implementasi Antarmuka Halaman Proses.....	92
5.3.4 Implementasi Antarmuka Halaman Hasil .....	94
5.3.5 Implementasi Antarmuka Halaman Perhitungan <i>Error</i> .....	95

BAB 6 PENGUJIAN DAN ANALISIS.....	96
6.1 Sistematika Pengujian .....	96
6.2 Hasil dan Analisis .....	96
6.2.1 Pengujian Pengaruh <i>Order</i> Terhadap MSE .....	96
6.2.2 Pengujian Pengaruh Data <i>Training</i> Terhadap MSE .....	98
6.2.3 Pengujian Validasi Sistem.....	99
BAB 7 PENUTUP .....	102
7.1 Kesimpulan.....	102
7.2 Saran.....	103
Daftar Pustaka .....	104
Lampiran.....	106
Lampiran 1 Daftar Historis Harga Cabai .....	106
Lampiran 2 Tabel Lengkap Hasil Pengujian Pengaruh <i>Order</i> terhadap MSE .....	108
Lampiran 3 Tabel Lengkap Hasil Pengujian Pengaruh Data <i>Training</i> terhadap MSE.....	110

## DAFTAR TABEL

Tabel 2.1 Kajian pustaka .....	8
Tabel 2.2 <i>Fuzzy logic relationship (FLR)</i> .....	21
Tabel 2.3 Peramalan <i>fuzzy set</i> .....	22
Tabel 3.1 Data historis harga cabai .....	27
Tabel 3.2 Rencana jadwal penelitian .....	28
Tabel 4.1 Data sampel perhitungan manual.....	50
Tabel 4.2 Hasil kuadrat dari selisih data ke- <i>i</i> dan <i>mean</i> .....	51
Tabel 4.3 Nilai minimum, nilai maksimum dan simpangan baku .....	52
Tabel 4.4 Hasil perhitungan <i>universe of discourse</i> .....	52
Tabel 4.5 Tabel nilai absolut selisih data dengan periode sebelumnya .....	53
Tabel 4.6 Hasil jumlah <i>cluster</i> data .....	54
Tabel 4.7 Matriks partisi awal pada <i>cluster</i> data A.....	55
Tabel 4.8 Derajat keanggotaan matriks partisi data A.....	56
Tabel 4.9 Perhitungan pusat <i>cluster</i> data A.....	56
Tabel 4.10 Fungsi objektif data A.....	57
Tabel 4.11 Normalisasi matriks partisi .....	58
Tabel 4.12 Hasil normalisasi matriks partisi.....	58
Tabel 4.13 Hasil pusat <i>cluster</i> .....	59
Tabel 4.14 Batas subinterval masing–masing atribut data.....	59
Tabel 4.15 Hasil derajat keanggotaan dan <i>fuzzy set</i> data A .....	61
Tabel 4.16 Hasil fuzzifikasi semua data atribut .....	62
Tabel 4.17 Hasil pembentukan FLR data <i>training</i> .....	63
Tabel 4.18 Hasil pembentukan FLR data <i>testing</i> .....	63
Tabel 4.19 Frekuensi FLR yang cocok dengan data <i>testing</i> .....	65
Tabel 4.20 Perbandingan data aktual dan hasil peramalan.....	65
Tabel 4.21 Keterangan rancangan antarmuka halaman input .....	67
Tabel 4.22 Keterangan rancangan antarmuka halaman tampil data .....	68
Tabel 4.23 Keterangan rancangan antarmuka halaman proses <i>clustering</i> .....	70
Tabel 4.24 Keterangan rancangan antarmuka proses fuzzifikasi .....	71
Tabel 4.25 Keterangan rancangan antarmuka proses <i>fuzzy logical relationship</i> .	72
Tabel 4.26 Keterangan rancangan antarmuka halaman hasil peramalan .....	73

Tabel 4.27 Keterangan rancangan antarmuka halaman perhitungan <i>error</i> .....	74
Tabel 4.28 Rancangan pengujian pengaruh <i>order</i> terhadap MSE .....	75
Tabel 4.29 Rancangan pengujian pengaruh data <i>training</i> terhadap MSE .....	76
Tabel 4.30 Rancangan pengujian validasi sistem .....	76
Tabel 5.1 Spesifikasi perangkat keras .....	77
Tabel 5.2 Spesifikasi perangkat lunak .....	77
Tabel 6.1 Hasil pengujian pengaruh <i>order</i> terhadap MSE .....	96
Tabel 6.2 Hasil pengujian pengaruh data <i>training</i> terhadap MSE .....	98
Tabel 6.3 Hasil pengujian validasi sistem .....	99
Tabel 6.4 Perbandingan data hasil peramalan.....	101



## DAFTAR GAMBAR

Gambar 3.1 Diagram alir metode high order <i>fuzzy times series multifactors</i> .....	25
Gambar 4.1 Diagram alir metode penelitian yang digunakan (bagian 1) .....	29
Gambar 4.2 Diagram alir metode penelitian yang digunakan (bagian 2) .....	30
Gambar 4.3 Diagram alir proses perhitungan <i>universe of discourse</i> .....	31
Gambar 4.4 Diagram alir proses perhitungan <i>mean</i> .....	32
Gambar 4.5 Diagram alir proses perhitungan simpangan baku .....	33
Gambar 4.6 Diagram alir proses perhitungan nilai <i>Dmin</i> dan <i>Dmax</i> .....	34
Gambar 4.7 Diagram alir proses perhitungan jumlah <i>cluster</i> .....	35
Gambar 4.8 Diagram alir proses perhitungan pusat <i>cluster</i> (bagian 1) .....	36
Gambar 4.9 Diagram alir proses perhitungan pusat <i>cluster</i> (bagian 2) .....	37
Gambar 4.10 Diagram alir proses perhitungan pusat <i>cluster</i> (bagian 3) .....	38
Gambar 4.11 Diagram alir proses perhitungan pusat <i>cluster</i> (bagian 4) .....	39
Gambar 4.12 Diagram alir proses perhitungan pusat <i>cluster</i> (bagian 5) .....	40
Gambar 4.13 Diagram alir pembentukan subinterval dan <i>fuzzy set</i> (bagian 1) ...	41
Gambar 4.14 Diagram alir pembentukan subinterval dan <i>fuzzy set</i> (bagian 2) ...	42
Gambar 4.15 Diagram alir proses fuzzifikasi (bagian 1) .....	43
Gambar 4.16 Diagram alir proses fuzzifikasi (bagian 2) .....	44
Gambar 4.17 Diagram alir proses pembentukan FLR (bagian 1) .....	45
Gambar 4.18 Diagram alir proses pembentukan FLR (bagian 2) .....	46
Gambar 4.19 Diagram alir proses pembentukan FLR (bagian 3) .....	47
Gambar 4.20 Diagram alir proses defuzzifikasi (bagian 1) .....	48
Gambar 4.21 Diagram alir proses defuzzifikasi (bagian 2) .....	49
Gambar 4.22 Grafik <i>fuzzy set</i> data A .....	61
Gambar 4.23 Rancangan antarmuka halaman input .....	66
Gambar 4.24 Rancangan antarmuka halaman tampil data .....	68
Gambar 4.25 Rancangan antarmuka halaman proses <i>clustering</i> .....	69
Gambar 4.26 Rancangan antarmuka proses fuzzifikasi .....	71
Gambar 4.27 Rancangan antarmuka proses <i>fuzzy logical relationship</i> (FLR) .....	72
Gambar 4.28 Rancangan antarmuka halaman hasil peramalan .....	73
Gambar 4.29 Rancangan antarmuka halaman perhitungan <i>error</i> .....	74
Gambar 5.1 Implementasi antarmuka halaman input .....	91

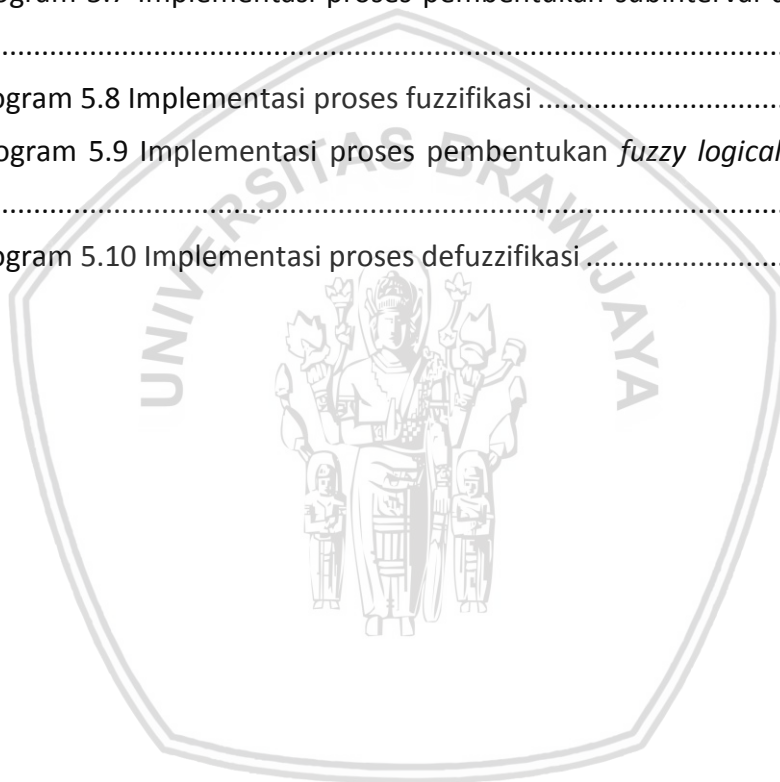


Gambar 5.2 Implementasi antarmuka halaman tampil data.....	92
Gambar 5.3 Implementasi antarmuka halaman proses <i>clustering</i> .....	92
Gambar 5.4 Implementasi antarmuka halaman proses fuzzifikasi.....	93
Gambar 5.5 Implementasi antarmuka halaman proses <i>bagian fuzzy logical relationship</i> .....	94
Gambar 5.6 Implementasi halaman antarmuka hasil.....	94
Gambar 5.7 Implementasi halaman antarmuka perhitungan <i>error</i> .....	95
Gambar 6.1 Grafik hasil pengujian pengaruh <i>order</i> terhadap MSE.....	97
Gambar 6.2 Grafik hasil pengujian pengaruh data <i>training</i> terhadap MSE.....	99
Gambar 6.3 Grafik hasil pengujian validasi sistem .....	100
Gambar 6.4 Grafik perbandingan data aktual dengan hasil peramalan .....	101



## DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi proses <i>universe of discourse</i> .....	78
Kode Program 5.2 Implementasi proses <i>mean</i> .....	79
Kode Program 5.3 Implementasi proses simpangan baku .....	79
Kode Program 5.4 Implementasi proses <i>Dmin</i> dan <i>Dmax</i> .....	80
Kode Program 5.5 Implementasi proses jumlah <i>cluster</i> .....	80
Kode Program 5.6 Implementasi proses pusat <i>cluster</i> .....	83
Kode Program 5.7 Implementasi proses pembentukan subinterval dan <i>fuzzy set</i> .....	84
Kode Program 5.8 Implementasi proses fuzzifikasi .....	86
Kode Program 5.9 Implementasi proses pembentukan <i>fuzzy logical relationship</i> .....	88
Kode Program 5.10 Implementasi proses defuzzifikasi .....	90



## DAFTAR LAMPIRAN

Lampiran 1 Daftar Historis Harga Cabai .....	106
Lampiran 2 Tabel Lengkap Pengujian Pengaruh <i>Order</i> terhadap MSE.....	108
Lampiran 3 Tabel Lengkap Pengujian Pengaruh Data <i>Training</i> terhadap MSE ..	110



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Komoditas pertanian memiliki peranan penting dalam menunjang perekonomian di Indonesia. Hal tersebut disebabkan komoditas pertanian merupakan mata pencaharian dan konsumsi utama masyarakat Indonesia. Kebutuhan sehari-hari masyarakat tidak lepas dari komoditas pertanian seperti cabai, bawang merah, bawang putih, tomat dan lain-lain. Kebutuhan ini terkait dengan kebutuhan konsumsi seperti digunakan untuk masakan, bumbu masakan dan lain-lain. Oleh karena itu, pemenuhan kebutuhan dari komoditas pertanian tersebut dapat dikatakan sebagai kebutuhan primer bagi masyarakat.

Dalam perdagangan, sebutan dari hasil pertanian yang termasuk dalam komoditas utama ini dibagi menjadi dua golongan besar. Golongan pertama adalah kelompok dengan harga yang cenderung stabil yang apabila terjadi kenaikan dan penurunan harga tidak begitu tajam. Kelompok sayuran dan hasil pertanian yang masuk dalam kelompok ini misalnya terong, mentimun, kacang panjang, kubis, wortel dan lain-lain. Golongan kedua adalah kelompok komoditas yang memiliki fluktuasi harga yang tajam. Komoditas yang masuk dalam kelompok ini misalnya cabai dan bawang merah. Ketika pasokan di pasar kurang, harganya dapat jauh melambung tinggi dari harga normal. Sebaliknya, saat pasokan berlebihan, harganya akan turun jauh di bawah harga normal.

Indonesia merupakan negara agraris yang dapat memproduksi sendiri sebagian besar kebutuhan bahan makanan pokoknya. Berdasarkan data dari Badan Pusat Statistik (BPS, 2015), produksi cabai segar dengan tangkai tahun 2014 adalah sebesar 1,075 juta ton. Dibandingkan pada tahun 2013, pada tahun 2014 terjadi kenaikan produksi sebesar 61,73 ribu ton (6,09 persen). Kenaikan ini disebabkan oleh kenaikan produktivitas sebesar 0,19 ton per hektar (2,33 persen) dan peningkatan luas panen sebesar 4,62 ribu hektar (3,73 persen) dibandingkan tahun sebelumnya. Sedangkan Produksi bawang merah tahun 2014 sebesar 1,234 juta ton. Dibandingkan tahun 2013, produksi meningkat sebesar 223,22 ribu ton (22,08 persen). Peningkatan ini disebabkan oleh meningkatnya luas panen sebesar 21,77 ribu hektar (22,00 persen) dibandingkan tahun 2013 (BPS, 2015).

Meskipun melimpahnya jumlah produksi, namun permintaan terhadap komoditi tersebut juga sangatlah tinggi. Dengan jumlah penduduk Indonesia yang mencapai 237.641.326 jiwa berdasarkan sensus pada tahun 2010 (BPS, 2016) dapat dipastikan bahwa kebutuhan akan komoditas pertanian akan sangat tinggi. Hal ini mengakibatkan harga dari komoditas tersebut menjadi tidak stabil dan

cenderung mengalami kenaikan. Kenaikan harga tersebut dipengaruhi berbagai faktor diantaranya adalah musim panen, jumlah pasokan, konsumsi masyarakat dan lain-lain (Farid, et al., 2012). Oleh karena itu diperlukan suatu metode untuk dapat memperkirakan harga dari komoditas strategis ini sehingga dapat digunakan untuk pendukung pengambilan keputusan terkait dengan permasalahan harga.

Peramalan adalah salah satu bidang dalam ilmu pengetahuan yang dapat membantu dalam pengambilan keputusan (*decision making*). Peramalan digunakan untuk melakukan prediksi suatu peristiwa yang akan terjadi di masa mendatang berdasarkan rekam jejak atau serangkaian peristiwa yang telah terjadi sebelumnya, seperti peramalan suhu, cuaca, saham, gempa bumi, dsb (Chen, et al., 2009). Berdasarkan hasil peramalan yang diperoleh, kita dapat mencegah terjadinya kerugian atau mendapat keuntungan dari aktivitas peramalan yang telah dilakukan. Faktanya, sebuah peristiwa dalam kehidupan nyata dapat dipengaruhi oleh banyak faktor. Semakin banyak fakta yang dipertimbangkan, maka semakin tinggi tingkat akurasi peramalan yang dihasilkan (Wang, et al., 2009).

Teknik peramalan ini memiliki berbagai metode yang memiliki kelebihan dan kekurangan tersendiri tergantung dengan jenis data yang akan diramalkan. Salah satu metode yang dapat digunakan sebagai peramalan adalah *Fuzzy Times Series*. Metode *Fuzzy Times Series* pertama kali diterapkan oleh Song dan Chissom (1993) untuk menyelesaikan permasalahan peramalan dengan cara melihat data sebelumnya untuk membentuk model *Fuzzy Times Series*. Salah satu penelitian menggunakan metode *Fuzzy Times Series* adalah peramalan pendaftar di Universitas Alabama dengan memodifikasi *Fuzzy Times Series* Stevenson dan Porter (Saxeena, et al., 2012). Metode *Fuzzy Times Series* yang digunakan dapat menghasilkan nilai *Average Forecasting Error Rate* (AFER) sebesar 0,34% dari hasil peramalan yang dilakukan.

Peramalan dengan menggunakan metode *Fuzzy Times Series* sudah terbukti menghasilkan tingkat kesalahan yang cukup rendah. Penelitian yang menggunakan metode *Fuzzy Times Series* selanjutnya adalah peramalan kebutuhan beban listrik harian di Malaysia (Efendi, et al., 2015). Kesimpulan dalam penelitian ini peramalan yang dilakukan menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) yang cukup rendah, yaitu antara 1,23% sampai 1,63% dengan data uji 5 sampai 10 data (Efendi, et al., 2015).

Terdapat beberapa faktor yang dapat memengaruhi tingkat akurasi peramalan dari metode *Fuzzy Times Series*, salah satunya adalah banyaknya variabel yang memengaruhi besarnya nilai hasil peramalan. Karakteristik harga cabai yang fluktuatif dan sangat dipengaruhi oleh berbagai faktor seperti jumlah



produksi cabai, jumlah konsumsi, harga cabai tingkat eceran serta luas lahan panen tanaman cabai dan lain-lain, maka diperlukan studi untuk mempelajari metode yang sesuai untuk mendapatkan hasil peramalan yang lebih baik.

Metode *High Order Fuzzy Times Series Multifactors* dapat digunakan untuk mengakomodasi beberapa variabel yang memengaruhi harga cabai seperti jumlah produksi, jumlah konsumsi, harga cabai dan luas lahan panen tanaman cabai. Keunggulan dari metode *Fuzzy Times Series Multifactors* dapat dilihat pada penelitian sebelumnya mengenai prediksi harga emas. Dengan beberapa faktor yang memengaruhi pasaran harga emas, penggunaan metode *High Order Fuzzy Times Series Multifactors* menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) sebesar 0,443% (Wardhani, 2015). Berdasarkan uraian tersebut dalam tugas akhir penulis akan mengambil judul Peramalan Harga Cabai menggunakan Metode *High Order Fuzzy Times Series Multifactors*. Penggunaan metode tersebut dalam peramalan diharapkan dapat membantu mengetahui perkiraan harga cabai dengan lebih akurat.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, rumusan masalah yang menjadi fokus utama dalam tugas akhir ini adalah:

1. Bagaimana pengaruh nilai *order* terhadap hasil peramalan harga cabai dengan menggunakan metode *High Order Fuzzy Times Series Multifactors*?
2. Bagaimana tingkat akurasi hasil peramalan harga cabai dengan menggunakan metode *High Order Fuzzy Times Series Multifactors*?

## 1.3 Tujuan

Tujuan dari penelitian peramalan harga cabai menggunakan metode *high order fuzzy times series multifactors* adalah:

1. Mengetahui pengaruh nilai *order* terhadap hasil peramalan harga cabai dengan menggunakan metode *High Order Fuzzy Times Series Multifactors*.
2. Mengetahui tingkat akurasi metode *High Order Fuzzy Times Series Multifactors* untuk meramalkan harga cabai.

## 1.4 Manfaat

Melalui tugas akhir ini diharapkan dapat memberikan manfaat yaitu:

Bagi Akademik

1. Memberikan kontribusi pengetahuan mengenai penerapan metode peramalan untuk membantu memprediksi harga cabai.

2. Menambah referensi dalam perumusan implementasi mengenai metode peramalan.

#### Bagi Umum

1. Memberikan gambaran bagi konsumen atau masyarakat umum untuk dapat mengetahui dan memantau harga komoditas pertanian dari periode waktu ke waktu.
2. Memberikan gambaran bagi pedagang dan petani tanaman cabai agar dapat mengetahui dan memantau harga cabai dan melakukan perencanaan penjualan dan produksi sesuai dengan informasi tersebut.
3. Informasi hasil peramalan juga dapat dijadikan patokan bagi pemangku kebijakan dalam rangka menerapkan harga cabai sehingga harga tetap stabil.

### 1.5 Batasan Masalah

Dalam penelitian Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors* ini digunakan batasan penelitian. Batasan penelitian tersebut digunakan agar penelitian ini dapat mencapai tujuan yang diharapkan pada bab sebelumnya. Batasan penelitian yang digunakan adalah:

1. Data yang digunakan yaitu data nasional terkait dengan harga dan faktor-faktor yang memengaruhi harga cabai dari tahun 2010–2015 dengan frekuensi perbulan.
2. Variabel yang dilibatkan dalam peramalan adalah jumlah produksi, permintaan/jumlah konsumsi, luas panen serta harga pada periode sebelumnya. Faktor lain yang memengaruhi harga mungkin tidak terakomodir dalam metode peramalan (misal inflasi, musim paceklik, hama, dll).
3. Metode yang digunakan dalam penelitian peramalan harga cabai adalah *High Order Fuzzy Times Series Multifactors*.

### 1.6 Sistematika Pembahasan

Untuk mencapai tujuan yang diuraikan di atas, maka sistematika pembahasan laporan penelitian Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors* adalah sebagai berikut:

#### BAB 1 PENDAHULUAN

Bab Pendahuluan berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan penelitian Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors*.

## **BAB 2 LANDASAN KEPUSTAKAAN**

Bab Landasan Kepustakaan menjelaskan kajian pustaka yang terkait dengan penelitian Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors*.

## **BAB 3 METODOLOGI PENELITIAN**

Bab Metodologi Penelitian menjelaskan tentang metode *High Order Fuzzy Times Series Multifactors* yang digunakan untuk melakukan peramalan harga cabai.

## **BAB 4 PERANCANGAN**

Bab Perancangan menjelaskan tentang bagaimana rancangan sistem peramalan harga cabai menggunakan metode *Fuzzy Times Series Multifactors*.

## **BAB 5 IMPLEMENTASI**

Bab Implementasi berisi tentang bagaimana implementasi peramalan harga cabai menggunakan metode *Fuzzy Times Series Multifactors*.

## **BAB 6 PENGUJIAN DAN ANALISIS**

Bab Pengujian dan Analisis menjelaskan tentang hasil pengujian pada sistem peramalan harga cabai menggunakan metode *Fuzzy Times Series Multifactors*.

## **BAB 7 PENUTUP**

Bab Penutup berisi kesimpulan yang diperoleh dari bab perancangan, implementasi, pengujian pada penelitian Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors* yang dikembangkan pada tugas akhir ini disertai saran–saran untuk pengembangan penelitian lebih lanjut.

## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang kajian pustaka, objek penelitian beserta metode yang digunakan pada penelitian Peramalan Harga Cabai Menggunakan Metode *High Order Fuzzy Times Series Multifactors*.

### 2.1 Kajian Pustaka

Pada penelitian yang telah ada sebelumnya, peramalan harga komoditas strategis pertanian cabai dilakukan dengan menggunakan metode *Backpropagation Neural Network*. Dalam penelitian tersebut peramalan harga cabai menggunakan parameter berupa jumlah produksi, jumlah permintaan, luas panen dan harga cabai pada periode sebelumnya. Metode *Backpropagation Neural Network* yang digunakan dalam peramalan tersebut menghasilkan tingkat akurasi yang didapatkan nilai MAPE (*Mean Absolute Percentage Error*) yaitu sebesar 16,193%. Kesimpulan yang diperoleh dari penelitian ini yaitu, metode ini dapat diterapkan untuk peramalan harga komoditas strategis tanaman cabai dengan tingkat kesalahan yang cukup besar yaitu 16,193% (Rofiq, 2017).

Penggunaan metode lain yang sering diterapkan untuk menyelesaikan masalah peramalan terkait data historis adalah *Fuzzy Times Series*. *Fuzzy Times Series* telah digunakan dalam penelitian yang dilakukan oleh Saxena, et al. (2012) untuk meramalkan jumlah pendaftar di Universitas Alabama. Dalam penelitian tersebut dilakukan upaya modifikasi terhadap metode *Fuzzy Times Series* yang dilakukan oleh Stevenson dan Porter (2009) dengan mengganti partisi interval berbasis kerapatan frekuensi dengan partisi interval berbasis rerata. Hasil penelitian tersebut menunjukkan bahwa peramalan yang dilakukan menghasilkan nilai *Average Forecasting Error Rate* (AFER) yang cukup rendah yaitu 0,34% (Saxeena, et al., 2012).

Metode *Fuzzy Times Series* telah terbukti menghasilkan peramalan dengan tingkat kesalahan yang cukup rendah. Penelitian dengan menggunakan metode *Fuzzy Times Series* juga digunakan untuk menentukan kebutuhan beban listrik harian di Malaysia (Efendi, et al., 2015). Pada penelitian tersebut menunjukkan bahwa peramalan yang dilakukan memiliki nilai *Mean Absolute Percentage Error* (MAPE) yang cukup rendah, yaitu antara 1,23% sampai 1,63% dengan data uji 5 sampai 10 data.

Dengan menggunakan metode *Fuzzy Times Series*, peramalan juga dilakukan untuk melakukan prediksi curah hujan (Irsyad, et al., 2013). Parameter yang digunakan pada penelitian tersebut merupakan data aktual curah hujan bulanan. Metode yang digunakan penelitian ini adalah *Automatic Clustering* dan *High Order*

*Fuzzy Times Series*. Pada penelitian ini metode *Automatic Clustering* digunakan untuk membangun interval dan *High Order Fuzzy Times Series* untuk melakukan peramalan curah hujan. Kesimpulan yang diperoleh dari penelitian ini yaitu, metode ini dapat diterapkan untuk peramalan curah hujan dan hasil evaluasi menggunakan metode *Mean Absolute Percentage Error* (MAPE) dari beberapa percobaan dapat diperoleh nilai kesalahan sebesar 4,37 % (Irsyad, et al., 2013).

Curah hujan di kota Samarinda juga menjadi objek penelitian selanjutnya. Inputan yang digunakan pada penelitian ini berupa data aktual bulanan curah hujan di kota Samarinda. Metode yang digunakan pada penelitian ini adalah *Fuzzy Times Series Chen* dengan pembentukan panjang interval berbasis rata-rata atau *Average Based*. Pada penelitian ini peramalan curah hujan terbaik yaitu dengan menggunakan data sebanyak 29 data dengan menghasilkan tingkat nilai kesalahan terkecil yaitu dengan metode *Root Mean of Squared Error* (RMSE) sebesar 73,68 dan metode *Mean Absolute Error* (MAE) sebesar 53,9. Pada penelitian ini disimpulkan bahwa data sebanyak 29 data memiliki tingkat kesalahan lebih kecil dibandingkan dengan menggunakan data dengan jumlah sampel yang lebih banyak (Fauziah, et al., 2016).

Pada penelitian selanjutnya objek penelitian yang diramalkan adalah harga emas harian. Masukan yang digunakan pada penelitian ini berupa data aktual harian harga emas indonesia, harga emas London, kurs Dollar ke Rupiah, dan kurs Euro ke Rupiah. Metode yang digunakan pada penelitian pada pembentukan sub interval menggunakan metode *Fuzzy C-means* dan untuk peramalan menggunakan metode *High Order Fuzzy Times Series Model*. Hasil pada penelitian ini, berdasarkan penggunaan metode evaluasi *Mean Absolute Percentage Error* (MAPE) dapat diperoleh nilai *error* terkecil sebesar 0,443%. Nilai *error* yang lebih kecil dari 10% ini menunjukkan bahwa metode ini sangat baik digunakan untuk peramalan harga emas dengan mempertimbangkan banyak faktor yang memengaruhinya (Wardhani, 2015).

Penelitian selanjutnya menggunakan objek berupa daging sapi. Daging sapi merupakan salah satu komoditas yang tingkat permintaannya selalu tinggi karena merupakan produk peternakan yang memiliki kandungan nilai gizi untuk pemenuhan kebutuhan protein bagi masyarakat. Permintaan daging sapi dipengaruhi oleh faktor produksi daging sapi, konsumsi daging sapi, dan tingkat pendapatan masyarakat. Dalam melakukan peramalan metode yang digunakan yaitu *Multifactors High Order Fuzzy Times Series Model*. Dalam penelitian ini diperoleh nilai *Average Forecasting Error Rate* (AFER) sebesar 6,648% yang menunjukkan bahwa nilai *error* yang semakin kecil berarti tingkat akurasi semakin



baik (Nugraha, 2017). Beberapa penelitian yang terkait ditunjukkan dalam Tabel 2.1.

**Tabel 2.1** Kajian pustaka

No	Judul	Objek	Metode	Perbedaan Skripsi Peneliti
1	Peramalan Komoditas Strategis Pertanian Cabai Menggunakan Metode <i>Backpropagation Neural Network</i> (Rofiq, 2017)	Peramalan Strategis Pertanian Cabai	<i>Backpropagation Neural Network</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>
2	<i>Forecasting Enrollments based on Fuzzy Time Series with Higher Forecast Accuracy Rate</i> (Saxeena, et al., 2012)	Peramalan Pendaftar di Universitas Alabama	<i>Fuzzy Times Series</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>
3	<i>A New Linguistic Out-Sample Approach of Fuzzy Time Series for Daily Forecasting of Malaysian Electricity Load Demand</i> (Efendi, et al., 2015)	Permintaan Beban Listrik Malaysia	<i>Fuzzy Times Series</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>

4	<i>Temperature Prediction and TAIEX Forecasting Based on Fuzzy Relationship and MTPSO techniques (Hsu, 2010)</i>	Peramalan Suhu Udara dan Taiwan Future Exchange (TAIFEX)	<i>Fuzzy Time Series dan Particle Swarm Optimization</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>
5	<i>Fuzzy Time Series Forecasting based on Fuzzy Logical Relationship and Similarity Measures (Cheng, et al., 2016)</i>	<i>Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)</i>	<i>Fuzzy Time Series, Fuzzy Logical Relationship, Particle Swarm Optimization, K-Means, dan Similarity Measure</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>
6	Peramalan Permintaan Daging Sapi Nasional Menggunakan Metode <i>Multifactors High Order Fuzzy Time Series Model</i> (Nugraha, 2017)	Peramalan Permintaan Daging Sapi Nasional	<i>Multifactors High Order Fuzzy Times Series Model</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>
7	Implementasi Metode <i>Multi-factors High order Fuzzy Time Series Model</i> untuk Prediksi Harga Saham (Wardhani, 2015)	Prediksi Harga Emas	<i>Multi-factors High order Fuzzy Times Series Model</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>

8	Peramalan Menggunakan <i>Fuzzy Time Series Chen</i> (Studi Kasus: Curah Hujan Kota Samarinda) (Fauziah, et al., 2016)	Curah Hujan Kota Samarinda	<i>Fuzzy Times Series Chen</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>
9	Penerapan Metode <i>Automatic Clustering</i> dan <i>High Order Fuzzy Time Series</i> Pada Peramalan Curah Hujan (Irsyad, et al., 2013)	Peramalan Curah Hujan	<i>High Order Fuzzy Times Series dan Automatic Clustering.</i>	Objek yang diteliti adalah peramalan harga cabai, metode yang digunakan adalah <i>High Order Fuzzy Times Series Multifactors</i>

## 2.2 Objek Penelitian

Pada bagian ini akan dijelaskan mengenai objek penelitian, yang mana objek penelitian ini berfungsi sebagai gambaran umum mengenai keterangan dan informasi mengenai komoditas tanaman yang akan dikaji dalam penelitian ini.

### 2.2.1 Cabai

Komoditas pertanian sendiri dapat diartikan sebagai hasil dari pertanian utama yang penting dan dibutuhkan oleh masyarakat untuk memenuhi kebutuhan sehari-hari. Permintaan pasar terhadap komoditi ini cenderung tinggi karena hasil pertaniannya dibutuhkan oleh hampir setiap lapisan masyarakat. Contoh dari hasil pertanian yang masuk dalam komoditi pertanian misalnya, cabai, bawang merah, bawang putih, dan lain-lain. Pada umumnya kebutuhan terhadap komoditi ini digunakan untuk memenuhi kebutuhan konsumsi seperti untuk makanan utama atau sebagai bumbu masakan. Selain itu, industri-industri besar juga memberikan permintaan yang tinggi terhadap komoditi ini, terlebih pada komoditi cabai.

Cabai digunakan sebagai penyedap untuk berbagai produk makanan olahan seperti mie instan dan sambal instan. Cabai merupakan komoditi penting dalam

perekonomian Indonesia. Dari segi konsumsi, memiliki pasar yang sangat signifikan yang dapat diketahui dari bobot inflasi yang mencapai 0,35% (BPS, 2011). Dari segi harga, cabai yang merupakan salah satu komoditi pertanian yang memiliki karakteristik yaitu sering terjadinya lonjakan harga yang sangat tinggi dan fluktuatif. Sebagai contoh, lonjakan harga di atas 100% kerap kali terjadi pada Desember tahun 2014 atau bahkan pada Januari 1996. Pada dua periode tersebut terjadi lonjakan harga sekitar 327% dibandingkan harga bulan sebelumnya (Farid, et al., 2012). Kenaikan harga cabai dapat berulang dalam 2–3 bulan dan kemudian terjadi penurunan setelahnya. Namun tidak jelas mengenai pola dari fluktuasi harga tersebut.

Salah satu hal yang turut memengaruhi harga komoditi cabai adalah stok cabai. Pasokan cabai yang tersedia sangat memengaruhi harganya di pasaran. Selain itu pola produksi dari petani yang secara tidak langsung memengaruhi jumlah posokan di pasar juga akan memengaruhi harga cabai. Ketika harga cabai naik di pasar, petani akan berbondong–bondong untuk menanam tanaman cabai. Namun, ketika masa panen cabai tiba justru pasokan di pasar telah melimpah ruah sehingga harga cabai bisa sangat turun drastis.

### 2.3 Peramalan

Terdapat beberapa definisi tentang peramalan menurut Kamus Besar Bahasa Indonesia (KBBI) yaitu:

- Peramalan merupakan prediksi, rencana atau estimasi kejadian masa depan yang tidak pasti (KBBI).
- Peramalan adalah sebagai penggunaan teknik–teknik statistik dalam bentuk gambaran masa depan berdasarkan pengolahan angka–angka yang terjadi pada masa lalu (Buffa, et al., 1996).
- Peramalan adalah bagian internal dari pengambilan keputusan manajemen (Makridakis, et al., 1999).

Berdasarkan beberapa definisi diatas dapat disimpulkan bahwa peramalan merupakan tindakan apa yang akan terjadi di masa mendatang, peramalan diperlukan untuk mengambil sebuah keputusan. Dalam pengambilan keputusan harus diambil berdasarkan pertimbangan yang ada. Oleh karena itu peramalan sangat penting hubungannya dengan pengambilan keputusan, sehingga kejadian di masa yang akan datang dapat dipersiapkan tindakan apa yang perlu dilakukan. Peramalan (*forecasting*) adalah proses pembuatan prediksi untuk nilai di masa depan dengan menggunakan data di masa lalu dan masa sekarang kemudian memproyeksikannya dengan model matematis. Peramalan sering juga diartikan dengan istilah perkiraan dan prediksi yang merujuk kepada metode–metode statistik.

Kegiatan peramalan merupakan bagian dari keputusan manajemen dengan mengurangi ketergantungan pada hal-hal yang belum pasti (intuisif). Untuk mendapatkan nilai prediksi dari suatu pola data, terdapat beberapa metode yang dapat diterapkan untuk melakukan peramalan. Tiap-tiap metode tersebut memiliki kelebihan dan kekurangannya masing-masing tergantung dengan jenis dan pola data yang ditangani. Kelebihan dan kekurangan ini terkait dengan tingkat akurasi yang akan dihasilkan dari peramalan tersebut.

Terdapat beberapa jenis pola data yang dapat dianalisis dengan metode-metode peramalan:

- Pola data konstan, yaitu data yang berfluktuasi di sekitar rata-rata secara stabil dan membentuk pola horisontal. Pola seperti ini muncul dalam jangka waktu yang pendek atau menengah. Oleh karena itu, jarang sekali suatu variabel memiliki pola konstan dalam jangka waktu yang panjang
- Pola data *trend*, yaitu apabila data pada jangka waktu tertentu memiliki kecenderungan meningkat dari waktu ke waktu maupun turun. Pola ini dapat disebabkan oleh beberapa faktor seperti bertambahnya populasi, perubahan pendapatan, dan pengaruh budaya
- Pola data *seasonal*, yaitu data yang memiliki pola gerakan yang berulang-ulang secara teratur dalam periode waktu tertentu, misalnya tahunan, bulanan dan sebagainya. Pola ini berhubungan dengan faktor iklim dan cuaca atau faktor yang dibuat oleh manusia seperti musim liburan dan lain-lain
- Pola data *cyclic*, yaitu data yang dipengaruhi oleh fluktuasi ekonomi jangka panjang, seperti alur hidup bisnis. Hal yang membedakan pola data *seasonal* dan *cyclic* adalah pola data *seasonal* memiliki panjang gelombang yang tetap dan terjadi dalam jarak waktu yang tetap, sedangkan pola *cyclic* memiliki durasi yang lebih panjang dan bervariasi
- Pola data residual atau variasi acak, yaitu data yang tidak teratur sama sekali. Data yang bersifat residu tidak dapat digambarkan

## 2.4 Deret Waktu (*Time Series*)

Dalam ilmu statistik, deret waktu (*time series*) adalah rangkaian data dari nilai pengamatan yang direkam dalam kurun waktu tertentu. Analisis deret waktu merupakan metode yang mempelajari deret waktu dari segi teori maupun diterapkan untuk membuat prediksi atau peramalan. Prediksi atau peramalan deret waktu adalah penggunaan model untuk memprediksi nilai di waktu mendatang berdasarkan rangkaian data historis (Siagian, et al., 2002).

Suatu data deret waktu terdiri dari urutan titik-titik data yang berturut-turut yang diukur pada suatu interval waktu yang kontinu, yang memiliki jarak antar dua



titik data yang sama dan untuk setiap satuan waktu dalam interval waktu hanya memiliki satu titik data. Analisis deret waktu meliputi metode-metode untuk menganalisis data deret waktu untuk mendapatkan nilai statistik yang bermakna dan karakteristik lain dari data tersebut. Peramalan data deret waktu adalah penggunaan model untuk memprediksi nilai masa depan berdasarkan nilai-nilai yang diamati sebelumnya. Sementara analisis regresi sering digunakan dengan cara seperti untuk menguji teori bahwa nilai-nilai saat ini dari satu atau lebih deret waktu independen memengaruhi nilai saat ini dari seri waktu lain, jenis analisis *time series* ini tidak disebut "analisis deret waktu", yang berfokus pada membandingkan nilai-nilai dari serangkaian waktu satu atau beberapa tergantung seri waktu di berbagai titik dalam waktu (Imdadullah, 2013).

## 2.5 Logika Fuzzy

Logika fuzzy merupakan salah satu komponen pembentuk *soft computing* yang pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965 dengan didasari teori himpunan fuzzy. Pada teori himpunan fuzzy, derajat keanggotaan berperan sangat penting sebagai penentu keberadaan elemen dalam suatu himpunan. Nilai keanggotaan, derajat keanggotaan atau *membership function* tersebutlah yang menjadi ciri utama dari penalaran menggunakan logika fuzzy (Kusumadewi, et al., 2013).

Beberapa alasan yang mendasari penggunaan logika fuzzy antara lain (Kusumadewi, et al., 2013):

- Konsep logika fuzzy mudah dimengerti karena berdasarkan pada teori himpunan.
- Logika fuzzy sangat fleksibel, mampu beradaptasi dengan adanya perubahan dan ketidakpastian dalam suatu permasalahan.
- Logika fuzzy memiliki toleransi terhadap keberadaan data yang tidak tepat.
- Logika fuzzy dapat memodelkan fungsi-fungsi non linier yang bersifat sangat kompleks.
- Logika fuzzy dapat membangun dan menerapkan pengalaman pakar secara langsung tanpa memerlukan proses pelatihan atau yang sering dikenal sebagai *Fuzzy Expert System*.
- Logika fuzzy dapat bekerja secara kooperatif dengan teknik-teknik kendali secara konvensional.
- Logika fuzzy didasarkan pada bahasa alami yang biasa digunakan sehari-hari sehingga mudah dimengerti.

Dalam memahami sistem fuzzy, terdapat beberapa hal yang perlu diketahui yaitu (Kusumadewi, et al., 2013):

a. Variabel *fuzzy*

Variabel *fuzzy* adalah variabel yang akan dibahas dalam suatu sistem *fuzzy*.

b. Himpunan *fuzzy*

Himpunan *fuzzy* adalah suatu kelompok yang merepresentasikan suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

c. Semesta pembicaraan

Semesta pembicaraan adalah seluruh nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy* yakni himpunan bilangan real yang selalu naik (bertambah) secara monoton dari kiri ke kanan.

d. Domain

Domain himpunan *fuzzy* adalah seluruh nilai yang diizinkan dalam semesta pembicaraan dan dapat dioperasikan dalam suatu himpunan *fuzzy*.

### 2.5.1 Himpunan Fuzzy

Pada teori himpunan klasik, nilai keanggotaan suatu entitas dalam suatu himpunan berdasarkan pada konsep fundamental dari sebuah himpunan yang mana sebuah individu merupakan anggota atau bukan anggota. Sebuah perbedaan yang jelas, tegas dan tidak ambigu terdapat pada anggota maupun bukan anggota dari sebuah himpunan yang didefinisikan dengan baik dalam teori ini, dan juga terdapat batasan yang sangat tepat dan jelas untuk menunjukkan bahwa suatu entitas terdapat pada suatu himpunan (Chen, et al., 2009). Nilai keanggotaan suatu entitas dalam suatu himpunan tegas (*crisp*) memiliki dua kemungkinan, yaitu satu (1) yang berarti bahwa suatu *item* menjadi anggota suatu himpunan, atau nol (0) yang berarti suatu entitas tidak menjadi anggota dari suatu himpunan (Kusumadewi, et al., 2013).

Teori himpunan *fuzzy* memiliki pendekatan yang berbeda dengan himpunan klasik. Sebuah himpunan *fuzzy* terdiri dari elemen-elemen yang memiliki beragam nilai keanggotaan dalam suatu himpunan, berbeda dengan himpunan klasik yang mengharuskan sebuah entitas memiliki nilai keanggotaan penuh untuk menjadi anggota dari sebuah himpunan (nilai keanggotaan bernilai satu). Berbeda dengan himpunan klasik, elemen pada himpunan *fuzzy* dapat menjadi anggota pada himpunan *fuzzy* lain pada semesta yang sama karena nilai keanggotaan elemen tersebut tidak harus penuh. Nilai keanggotaan elemen tersebut dihitung melalui fungsi yang menghitung kesamaran dan ambiguitas dalam suatu himpunan. Fungsi tersebut memetakan elemen himpunan *fuzzy* pada bilangan asli dengan interval 0 sampai 1 (Ross, 2010). Jika elemen  $x$  memiliki nilai keanggotaan *fuzzy*  $\mu_A(x) = 0$ , maka menunjukkan bahwa  $x$  bukan merupakan anggota himpunan  $A$  karena nilai keanggotaan dalam himpunan  $A$  adalah 0, demikian pula jika elemen  $x$  memiliki nilai keanggotaan *fuzzy*  $\mu_A(x) = 1$ , maka menunjukkan bahwa  $x$  merupakan

anggota penuh pada himpunan  $A$  karena nilai keanggotaan dalam himpunan  $x$  adalah 1.

Terkadang kerancuan muncul karena adanya kemiripan antara keanggotaan *fuzzy* dengan probabilitas. Keduanya memiliki nilai pada rentang 0 sampai 1, namun kedua kasus tersebut memiliki penafsiran yang sangatlah berbeda. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas menunjukkan proporsi terhadap frekuensi suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan suatu elemen adalah 0,95 pada himpunan *fuzzy* MUDA, maka berapa seringnya nilai itu diulang untuk memperoleh hasil yang hampir pasti MUDA tidak perlu dipermasalahkan. Di sisi lain, nilai probabilitas 0,95 MUDA berarti 5% elemen dari himpunan tersebut diharapkan tidak MUDA (Kusumadewi, et al., 2013). Himpunan *fuzzy* memiliki 2 atribut, yaitu (Kusumadewi, et al., 2013):

- a. Linguistik, yaitu penamaan suatu kelompok yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: TERANG, SAMAR, GELAP.
- b. Numeris, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel seperti: 15, 30, 60, dsb.

## 2.6 Fuzzy C-Means

*Fuzzy C-means* (FCM) pertama kali diperkenalkan oleh Dunn dan dikembangkan oleh Bezdek yang banyak digunakan dalam pengenalan pola. Metode ini merupakan pengembangan dari metode non-hierarkhi *K-Means Cluster*, selanjutnya dilakukan iterasi sampai mendapatkan keanggotaan kelompok tersebut.

FCM merupakan salah satu algoritma *fuzzy clustering* dan juga teknik klasterisasi dimana tiap data ditentukan oleh derajat keanggotaannya. Tujuan penggunaan algoritma FCM yaitu untuk menemukan pusat *cluster* atau *centroid* dengan meminimumkan fungsi objektif. Ouput dari FCM bukan merupakan *fuzzy inference system*, melainkan deretan pusat *cluster* dan beberapa derajat keanggotaan untuk setiap titik data. Hasil *output* ini dapat digunakan untuk membangun suatu *fuzzy inference system*. Berikut adalah langkah-langkah dari FCM (Kusumadewi, et al., 2013).

1. Langkah pertama, menentukan jumlah *cluster* ( $c \geq 2$ ), pembobot ( $\infty > w > 1$ ), maksimum iterasi (*maxIter*), *error* minimum yang diharapkan yaitu nilai positif yang sangat kecil ( $\xi$ ), fungsi objektif awal ( $P_0 = 0$ ), iterasi awal dideklarasikan 1.

- Langkah kedua, memasukkan data  $X$  yang akan diklasterisasi berupa matriks berukuran  $m \times n$  ( $m$  = jumlah sampel data,  $n$  = atribut setiap data). Contoh:  $X_{ij}$  = data sampel ke- $i$  ( $i = 1,2,3,4,\dots,m$ ), atribut ke- $j$  ( $j = 1,2,3,4,\dots,n$ ).
- Langkah ketiga, yaitu dengan membuat bilangan random  $\mu_{ik}$ ;  $i = 1,2,3,4,\dots,n$  dan  $k = 1,2,3,\dots,c$  ( $n$  = jumlah sampel pada data,  $c$  = jumlah *cluster* yang akan dibentuk) sebagai elemen-elemen matriks partisi awal. Untuk bilangan penyusun matriks partisi awal dihitung dijumlahkan setiap kolom yang ditunjukkan pada Persamaan 2.1.

$$Q_k = \sum_{k=1}^c \mu_{ik} \quad (2.1)$$

Keterangan:

$Q_k$  = jumlah bilangan pada kolom  $k$ ,

$\mu_{ik}$  = bilangan random pada baris  $i$ , kolom  $k$ ,

$c$  = jumlah kolom.

Selanjutnya untuk menghitung derajat keanggotaan awal dengan melakukan normalisasi pada matriks partisi, pada normalisasi dilakukan menggunakan Persamaan 2.2.

$$\mu_{ik} = \frac{\mu_{ik}}{Q_k} \quad (2.2)$$

- Langkah keempat, melakukan perhitungan *cluster* ke- $k$ :  $V_{kj}$ , dengan  $k = 1,2,3,4,\dots,c$  dan  $j = 1,2,3,4,\dots,m$ . Perhitungan pusat *cluster* ditunjukkan pada Persamaan 2.3.

$$V_{kj} = \frac{\sum_{i=1}^n ((u_{ik})^w \times X_{ij})}{\sum_{i=1}^n (u_{ik})^w} \quad (2.3)$$

Keterangan:

$V_{kj}$  = pusat *cluster* pada *cluster* ke- $k$  dan atribut ke- $j$ ,

$\mu_{ik}$  = derajat keanggotaan pada data ke- $i$ , *cluster* ke- $k$ ,

$X_{ij}$  = data sampel pada data ke- $i$ , atribut ke- $j$ ,

$w$  = pembobot.

- Langkah kelima, melakukan perhitungan fungsi objektif pada iterasi ke- $t$ :  $P_t$ . Perhitungan fungsi objektif ditunjukkan pada Persamaan 2.4.

$$P_t = \sum_{i=1}^n \sum_{k=1}^c ([\sum_{j=1}^m (X_{ij} - V_{kj})^2] (\mu_{ik})^w) \quad (2.4)$$

Keterangan:

$P_t$  = fungsi objektif,

$X_{ij}$  = data sampel ke- $i$ , atribut ke- $j$ ,

$V_{kj}$  = pusat *cluster* pada *cluster* ke- $k$  dan atribut ke- $j$ .

6. Langkah keenam, melakukan perhitungan normalisasi matriks partisi yang telah ditunjukkan Persamaan 2.5.

$$\mu_{ik} = \frac{\left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \quad (2.5)$$

Keterangan:

$X_{ij}$  = data sampel ke-i, atribut ke-j,

$V_{kj}$  = pusat *cluster* pada *cluster* ke-k dan atribut ke-j,

$w$  = pembobot.

7. Langkah ketujuh, yaitu melakukan pemeriksaan kondisi berhenti dengan kondisi sebagai berikut:
- Jika  $(|P_t - P_{t-1}| < \xi)$  atau  $(t > \text{MaxIter})$  maka berhenti,
  - Jika tidak memenuhi:  $t = t + 1$ , ulangi mulai dari langkah ke-4.

## 2.7 Fuzzy Times Series

Konsep *fuzzy times series* dikenalkan oleh Song dan Chissom pertama kali pada tahun 1993, yang mana nilai-nilainya direpresentasikan dengan himpunan *fuzzy* (Wang, et al., 2009). Sebuah himpunan *fuzzy*  $A$  dari semesta pembicaraan (*universe of discourse*)  $U$ , dalam hal ini  $U = \{u_1, u_2, \dots, u_n\}$ , didefinisikan pada Persamaan 2.6.

$$A = \frac{f_A(u_1)}{u_1} + \frac{f_A(u_2)}{u_2} + \dots + \frac{f_A(u_n)}{u_n} \quad (2.6)$$

dimana  $f_A$  merupakan fungsi keanggotaan dari himpunan *fuzzy*  $A$  pada Persamaan 2.7,

$$f_A: U \rightarrow [0,1] \quad (2.7)$$

menunjukkan derajat keanggotaan dari  $u_1$  dalam himpunan *fuzzy*, dan  $1 \leq i \leq n$ .

Jika  $Y(t) (t = \dots, 0, 1, 2, \dots)$  dianggap menjadi semesta pembicaraan yang mana himpunan *fuzzy* pada Persamaan 2.6 didefinisikan, lalu  $F(t)$  merupakan kumpulan dari  $f_i(t) (i = 1, 2, \dots)$ , maka  $F(t)$  disebut sebagai *fuzzy times series* pada Persamaan 2.8.

$$f_i(t) (i = 1, 2, \dots) \quad (2.8)$$

$$Y(t) (t = \dots, 0, 1, 2, \dots) \quad (2.9)$$

Jika terdapat relasi *fuzzy*  $(t, t-1)$  di antara  $F(t-1)$  dan  $F(t)$ , sebagaimana Persamaan 2.10.

$$F(t) = F(t-1) \circ R(t, t-1), \quad (2.10)$$



Yang mana “ $\circ$ ” adalah operator komposisi Max–Min, maka  $F(t)$  dianggap disebabkan oleh  $F(t-1)$ , dimana  $F(t)$  dan  $F(t-1)$  merupakan himpunan fuzzy.

Jika  $F(t)$  dianggap sebagai sebuah *fuzzy times series*, lalu  $F(t)$  digambarkan pada Persamaan 2.11.

$$F(t-1), F(t-2), \dots, F(t-n), \quad (2.11)$$

Sehingga *Fuzzy Logical Relationship* dapat direpresentasikan dengan sebuah *High-Order Fuzzy Logical Relationship* yang dinotasikan pada Persamaan 2.12.

$$F(t-n), \dots, F(t-1) \rightarrow F(t), \quad (2.12)$$

$$F(t-n), \dots, F(t-2), F(t-1), F(t) \quad (2.13)$$

Persamaan 2.13 merupakan himpunan fuzzy, dan disebut sebagai *model th-order fuzzy times series*. Jika  $F1(t)$  dan  $F2(t)$  dianggap sebagai dua buah *fuzzy times series*, lalu Persamaan 2.14 disebabkan oleh Persamaan 2.15.

$$(F1(t), F2(t)) \quad (2.14)$$

$$(F1(t-1), F2(t-1)), (F1(t-2), F2(t-2)), \dots, (F1(t-n), F2(t-n)) \quad (2.15)$$

maka *fuzzy logical relationship* dapat direpresentasikan dengan Persamaan 2.16.

$$(F1(t-n), F2(t-n)), \dots, (F1(t-1), F2(t-1)) \rightarrow (F1(t), F2(t)) \quad (2.16)$$

dan disebut sebagai *model two-factor th-order fuzzy times series*, dimana  $F1(t)$  dan  $F2(t)$  disebut sebagai *fuzzy times series “main factor”* dan *“second factor”*, masing–masing ( $t = \dots, 0, 1, 2, \dots$ ) (Wang & Chen, 2009).

## 2.8 Peramalan dengan *High Order Fuzzy Times Series Multifactors*

Metode *High Order Fuzzy Times Series Multifactors* adalah metode *fuzzy times series* yang melibatkan beberapa faktor yang memengaruhi peramalan dengan berdasarkan beberapa periode waktu sebelum periode yang akan diramalkan dalam proses pembentukan *Fuzzy Logical Relationship* (FLR). Hasil peramalan dengan banyak faktor yang memengaruhi dinilai lebih baik daripada dengan hanya satu faktor yang memengaruhi (Lin, et al., 2009). Pada metode ini pembentukan subinterval yang berfungsi dalam pendefinisian *fuzzy set* dilakukan dengan menggunakan teknik *Fuzzy C-means*. Berikut adalah langkah–langkah *Metode High Order Fuzzy Times Series Multifactors* (Lin, et al., 2009).

**Langkah 1:** Menentukan semesta pembicaraan (*Universe of discourse*).

Sebelum menghitung nilai U, terlebih dulu menghitung nilai *mean* (rata–rata) sampel kemudian menghitung nilai simpangan baku ( $\sigma$ ). *Mean* merupakan



perhitungan jumlah dari semua nilai dalam kumpulan data kemudian dibagi dengan jumlah total data. Untuk menghitung nilai mean ditunjukkan pada Persamaan 2.17. Sedangkan simpangan baku merupakan akar kuadrat dari *varians*, dan juga merupakan ukuran keragaman data statistik untuk menentukan bagaimana sebaran data dalam sampel. Untuk menghitung nilai simpangan baku ( $\sigma$ ) ditunjukkan pada Persamaan 2.18. Perhitungan U ditunjukkan pada Persamaan 2.19.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.17)$$

Keterangan:

$\bar{x}$  = *mean* (rata-rata),

$x_i$  = nilai data ke-i,

$n$  = banyak data.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (2.18)$$

Keterangan:

$\sigma$  = Simpangan baku data.

$$U = [D_{min} - \sigma, D_{max} + \sigma] \quad (2.19)$$

Keterangan:

$U$  = *Universe Of Discourse*,

$D_{min}$  = nilai minimum dari data sampel,

$D_{max}$  = nilai maksimum dari data sampel.

**Langkah 2:** Menentukan jumlah *cluster* data.

Proses klasterisasi bertujuan untuk membagi U ke dalam beberapa subinterval. Hasil perhitungan banyaknya *cluster* apabila bernilai pecahan maka harus dibulatkan menjadi bilangan bulat. Perhitungan banyaknya *cluster* ditunjukkan pada Persamaan 2.20.

$$k = \left\lceil |D_{min} - D_{max}| / \frac{\sum_{t=1}^n |X(t) - X(t-1)|}{n-1} \right\rceil \quad (2.20)$$

Keterangan:

$k$  = banyaknya / jumlah *cluster*,

$D_{min}$  = nilai minimum dari data sampel,

$D_{max}$  = nilai maksimum dari data sampel,

$n$  = banyaknya data sampel,

$X(t)$  = data pada waktu ke-t.

**Langkah 3:** Melakukan pembentukan pusat *cluster*.

Pada metode ini proses klasterisasi ditentukan untuk mendapatkan pusat *cluster* sebanyak  $k$ . Tahapannya,  $U$  dibagi kedalam  $k$  subinterval  $(D_{min}, d_1), (d_1, d_2), (d_2, d_3), \dots, (d_{k-1}, D_{max})$ , yang mana  $d_i$  ( $i = 1, 2, 3, \dots, k-1$ ) merupakan titik tengah antara dua pusat *cluster*.  $k$  subinterval didefinisikan sebagai  $u_1, u_2, u_3, \dots, u_k$ . Sehingga dalam hal ini subinterval dengan panjang berbeda dapat menggambarkan struktur data kompleks dengan baik dan dapat meningkatkan peramalan dibandingkan menggunakan subinterval dengan panjang yang sama.

**Langkah 4:** Melakukan pembentukan himpunan *fuzzy* atau *fuzzy set*.

Pada langkah ini *fuzzy set* direpresentasikan dengan kurva bahu. Pembentukan *fuzzy set* ditunjukkan pada Persamaan 2.21.

$$\begin{aligned} A_1 &= \frac{f_{11}}{u_1} + \frac{f_{12}}{u_2} + \frac{f_{13}}{u_3} + \dots + \frac{f_{1k}}{u_k} \\ &\dots = \dots + \dots + \dots + \dots + \dots \\ A_k &= \frac{f_{k1}}{u_1} + \frac{f_{k2}}{u_2} + \frac{f_{k3}}{u_3} + \dots + \frac{f_{kk}}{u_k} \end{aligned} \quad (2.21)$$

**Langkah 5:** proses fuzzifikasi.

Proses fuzzifikasi bertujuan untuk mengubah data sampel yang masih berupa bilangan crisp sehingga menjadi bilangan *fuzzy*. Masing-masing data dihitung derajat keanggotaannya terhadap *fuzzy set* yang sudah terbentuk, selanjutnya derajat keanggotaan dihitung berdasarkan kurva bahu dan derajat keanggotaan tertinggi menunjukkan hasil fuzzifikasi data tersebut. Contohnya  $\{A(t)\}$ ,  $\{B(t)\}$ ,  $\{C(t)\}$ , dan  $\{D(t)\}$  dimana  $t = 1, 2, 3, \dots, n$  merupakan *time series* dari faktor yang memengaruhi serta  $n$  merupakan banyaknya data. Berdasarkan pembagian subinterval masing-masing, *time series* ini difuzzifikasikan. Misalkan hasil dari Persamaan 2.22 yaitu menghitung banyak *cluster* dari  $k_1, k_2, k_3, k_4$ . Jadi *time series* difuzzifikasikan menjadi empat *fuzzy times series* yaitu:

$$\begin{aligned} \{A(t, i)\}, \text{ dimana } i &= 1, 2, 3, 4, \dots, k_1, \\ \{B(t, j)\}, \text{ dimana } j &= 1, 2, 3, 4, \dots, k_2, \\ \{C(t, p)\}, \text{ dimana } p &= 1, 2, 3, 4, \dots, k_3, \\ \{D(t, q)\}, \text{ dimana } q &= 1, 2, 3, 4, \dots, k_4, t = 1, 2, 3, \dots, n. \end{aligned} \quad (2.22)$$

**Langkah 6:** Melakukan pembentukan *Fuzzy Logical Relationship* (FLR).

Data sampel sebanyak  $n$  periode dipilih sebagai data *training* untuk membentuk *fuzzy times series* model. Misalkan  $t$  merupakan periode waktu yang akan diramal dengan menggunakan *order*-3, maka tiga periode sebelum periode  $t$  adalah  $t-3, t-$

2, dan  $t-1$  dimana  $t = 4,5,6,...,n$ . Maka FLR yang dihasilkan ditunjukkan pada Persamaan 2.23 dan berdasarkan persamaan tersebut maka akan membentuk sebuah kumpulan FLR dari data *training*.

$$(A(t-3,i_3), B(t-3,j_3), C(t-3,p_3), D(t-3,q_3)), \quad (2.23)$$

$$(A(t-2,i_2), B(t-2,j_2), C(t-2,p_2), D(t-2,q_2)),$$

$$(A(t-1,i_1), B(t-1,j_1), C(t-1,p_1), D(t-1,q_1)) \rightarrow A(t,i_4)$$

**Tabel 2.2** Fuzzy logic relationship (FLR)

$(A(t-3,i_3), B(t-3,j_3), C(t-3,p_3), D(t-3,q_3)),$
$(A(t-2,i_2), B(t-2,j_2), C(t-2,p_2), D(t-2,q_2)),$
$(A(t-1,i_1), B(t-1,j_1), C(t-1,p_1), D(t-1,q_1)) \rightarrow A(t,i_4)$
$(A(t-2,i_3), B(t-2,j_3), C(t-2,p_3), D(t-2,q_3)),$
$(A(t-1,i_2), B(t-1,j_2), C(t-1,p_2), D(t-2,q_2)),$
$(A(t,i_1), B(t,j_1), C(t,p_1), D(t,q_1)) \rightarrow A(t+1,i_4)$
$\vdots$
$(A(t+n-7,i_3), B(t+n-7,j_3), C(t+n-7,p_3), D(t+n-7,q_3)),$
$(A(t+n-6,i_2), B(t+n-6,j_2), C(t+n-6,p_2), D(t+n-6,q_2)),$
$(A(t+n-5,i_1), B(t+n-5,j_1), C(t+n-5,p_1), D(t+n-5,q_1)) \rightarrow A(t+n-4,i_4)$

Sumber: (Lin, et al., 2009).

#### Langkah 7: Proses Defuzzifikasi.

Berdasarkan langkah sebelumnya maka dapat diambil hasil *fuzzy* ke-H yaitu yang digunakan sebagai data *testing* atau data uji seperti yang ditunjukkan pada Persamaan 2.24.

$$(A(H-3,i_3), B(H-3,j_3), C(H-3,p_3), D(H-3,q_3)), \quad (2.24)$$

$$(A(H-2,i_2), B(H-2,j_2), C(H-2,p_2), D(H-2,q_2)),$$

$$(A(H-1,i_1), B(H-1,j_1), C(H-1,p_1), D(H-1,q_1)) \rightarrow A(H,i_4)$$

Faktor–faktor di ruas kiri pada Persaman 2.29 dibandingkan satu per satu dengan FLR data *training* yang sudah dicatat sebelumnya pada Tabel 2.2, membandingkannya yaitu dengan cara menghitung nilai absolut dari selisih antara pangkat bawah kedua faktor pada posisi yang sama di FLR. Terdapat 12 *fuzzy set* di ruas kiri pada masing–masing FLR sehingga dari hasil perbandingan diperoleh 12 nilai absolut yang kemudian dijumlahkan untuk mendapat total selisihnya. Jadi jika total selisih lebih kecil dari *threshold* yang ditentukan, maka FLR tersebut dinyatakan cocok.

Selanjutnya jika FLR menemukan N pasangan yang cocok pada Tabel 2.2, maka N faktor di ruas kanan dari N FLR ini mendasari peramalan *fuzzy set*  $A(*,*)$  yang mana jumlah dari setiap *fuzzy set* pada umumnya berbeda. Untuk peramalan dengan *fuzzy set* ditunjukkan pada Tabel 2.3.

**Tabel 2.3** Peramalan *fuzzy set*

Ruas Kanan	$A(*,1)$	$A(*,2)$	$A(*,3)$	....	$A(*,k1)$
Pusat <i>cluster</i>	$c_1$	$c_2$	$c_3$	....	$ck_1$
Frekuensi	$f_1$	$f_2$	$f_3$	....	$fk_1$

Sumber (Lin, et al., 2009)

Berdasarkan Tabel 2.3, untuk melakukan proses defuzzifikasi atau mengubah bilangan *fuzzy* menjadi bilangan *crisp* adalah menggunakan metode *centroid*. Proses defuzzifikasi dapat ditunjukkan pada Persamaan 2.25.

$$r_H = \frac{\sum_{i=1}^{k_1} c_i \times f_i}{\sum_{i=1}^{k_1} f_i} \quad (2.25)$$

Keterangan:

$r_H$  = hasil (*result*) peramalan data ke-H,

$c_i$  = pusat *cluster*  $i$   $A(*,i)$  adalah ruas kanan dari FLR hasil proses pencocokan,

$f_i$  = frekuensi / banyaknya muncul  $A(*,i)$  pada saat proses pencocokan.

## 2.9 Perhitungan Keakuratan

Dalam melakukan perencanaan kegiatan peramalan tentu menginginkan hasil ramalan yang tepat atau paling tidak hampir mendekati nilai aktualnya. Sehingga rencana yang dibuat menjadi rencana yang realistis. Keakuratan atau ketepatan merupakan yang menjadi kriteria *performance* suatu metode peramalan. Keakuratan tersebut dapat dinyatakan sebagai kesalahan peramalan (*forecast error*). Semakin kecil nilai kesalahan peramalan maka semakin tinggi keakuratan metode peramalan yang digunakan, dan sebaliknya semakin tinggi nilai kesalahan maka semakin rendah keakuratan metode peramalan yang digunakan.

*Mean Square Error* (MSE) adalah salah satu metode yang digunakan untuk menghitung nilai akurasi atau keakuratan dari metode yang digunakan sebagai peramalan atau prediksi. MSE merupakan perhitungan pangkat dua dari selisih antara nilai data hasil peramalan dengan nilai data aktual dan kemudian dibagi dengan banyaknya data yang telah diramalkan. Pada penentuan nilai evaluasi memilih menggunakan metode MSE karena metode ini dapat menunjukkan hasil skala kemiripan antara hasil data peramalan dengan data aktual. Berikut merupakan metode *Mean Square Error* (MSE) yang ditunjukkan pada Persamaan 2.26. (Wang, et al., 2009).

$$MSE = \frac{\sum_{i=1}^n (F_t - A_t)^2}{n} \quad (2.26)$$

Keterangan:

$MSE$  = nilai *Mean Square Error Rate*,

$A_t$  = nilai aktual pada data ke- $t$ ,

$F_t$  = nilai hasil peramalan untuk data ke- $t$ ,

$n$  = banyaknya data.

## BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai metodologi yang digunakan dalam pengerjaan penelitian. Metodologi yang digunakan dalam melaksanakan penelitian merupakan dasar penyusunan rencana penelitian dan merupakan gambaran dari penelitian secara umum. Metodologi penelitian digunakan agar penelitian yang dilakukan dapat mencapai tujuan secara efektif dan efisien.

### 3.1 Tipe Penelitian

Dalam Penelitian ini, pendekatan yang digunakan adalah penelitian nonimplementatif. Dalam pelaksanaannya permasalahan yang menjadi topik utama dalam penelitian ini yaitu permasalahan harga komoditas cabai yang selalu fluktuatif. Untuk itu penelitian ini bertujuan untuk dapat melakukan peramalan harga komoditas cabai di tingkat nasional berdasarkan data–data historis cabai dari waktu ke waktu.

Metode/teknik yang digunakan penelitian ini adalah Studi Kasus (*case study*). Dalam penelitian ini, peneliti menyelidiki secara cermat suatu rangkaian peristiwa perubahan data historis harga cabai yang dipengaruhi oleh beberapa faktor, seperti harga, jumlah produksi, jumlah konsumsi dan luas lahan pertanian cabai. Peneliti mengumpulkan informasi secara lengkap dengan menggunakan berbagai prosedur pengumpulan data berdasarkan kurun waktu yang ditentukan yakni antara tahun 2010–2015.

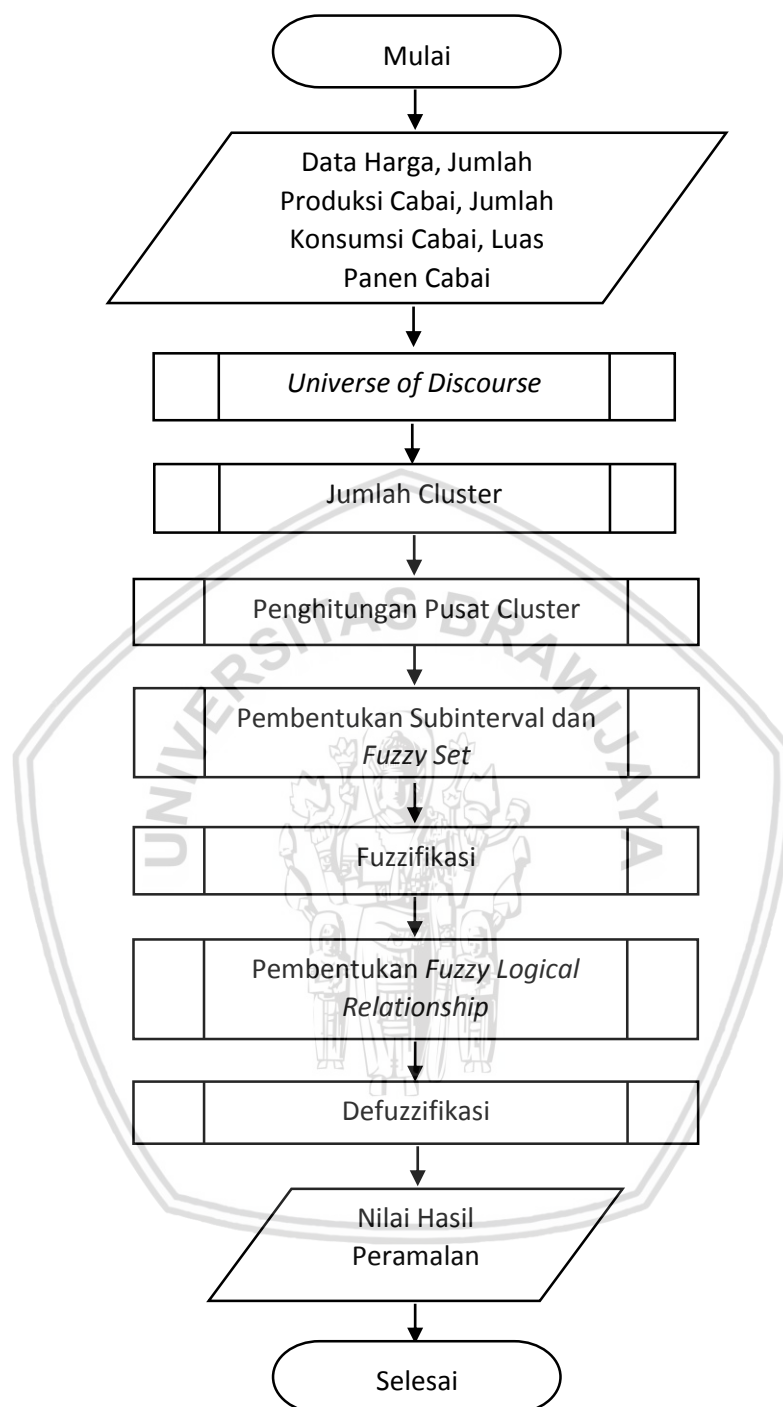
### 3.2 Strategi dan Rancangan Penelitian

Strategi dan Rancangan penelitian merupakan tahapan dalam pelaksanaan penelitian yang harus dilakukan agar penelitian yang dilakukan dapat mencapai tujuan.

#### 3.2.1 Gambaran Umum Sistem

Langkah selanjutnya setelah menentukan cara-cara untuk memperoleh data penelitian adalah memberikan gambaran umum sistem. Gambaran umum sistem merupakan gambaran sistem secara keseluruhan dalam penelitian peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors*. Gambaran umum sistem peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* ditunjukkan dalam bentuk diagram alir pada Gambar 3.1





**Gambar 3.1** Diagram alir metode *high order fuzzy times series multifactors*

### 3.2.2 Lokasi Penelitian

Dalam penulisan skripsi ini penulis melakukan penelitian di Fakultas Ilmu Komputer, Universitas Brawijaya Malang. Penelitian dilakukan dengan menggali informasi terkait dengan teori–teori, kajian dan referensi yang berhubungan dengan judul penelitian yaitu peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors*.

### 3.2.3 Pengumpulan Data

Sesuai dengan tujuan yang ingin dicapai dalam penelitian ini, penulis mengumpulkan data sebagai bahan utama penulisan laporan. Teknik pengumpulan data yang digunakan yaitu studi dokumentasi. Studi dokumentasi merupakan teknik pengumpulan data yang tidak langsung ditujukan kepada subjek penelitian, melainkan kepada dokumen-dokumen tertentu. Terdapat dua macam sumber data yang digunakan dalam teknik studi dokumentasi, yakni sumber data primer dan sekunder. Sumber data primer merupakan data atau dokumen yang ditulis oleh orang yang secara langsung mengalami suatu peristiwa. Sedangkan sumber data sekunder adalah dokumen yang ditulis oleh orang lain yang mendapat cerita dari pelaku peristiwa.

Data yang akan digunakan oleh penulis dalam penelitian ini berupa data sekunder. Sumber data sekunder ini terkait dengan harga dan faktor–faktor yang memengaruhi harga komoditas cabai. Data yang digunakan merupakan data pada periode tahun 2010–2015 tingkat nasional dengan frekuensi bulanan yang diadaptasi dari Badan Pusat Statistik (BPS). Data–data yang dikumpulkan adalah data berikut:

- a. Harga konsumen tingkat eceran cabai merah  
Berupa data harga cabai merah konsumen di Indonesia pada tingkat eceran selama tahun 2010 hingga 2015 dengan satuan rupiah. Harga cabai merah yang digunakan berupa data nasional.
- b. Jumlah produksi cabai merah  
Berupa data total produksi cabai merah di Indonesia pada tahun 2010 hingga 2015 dengan satuan ton. Data produksi yang digunakan adalah data nasional.
- c. Konsumsi cabai merah  
Berupa data konsumsi perkapita cabai merah dari tahun 2010 hingga tahun 2015. Data konsumsi ini merupakan data mikro sehingga diperlukan permintaan khusus kepada pihak BPS. Data konsumsi cabai merah yang digunakan juga merupakan data nasional.

d. Luas panen cabai merah

Berupa data daerah luas panen cabai merah pada tahun 2010 hingga 2015 dengan satuan hektar. Data luas panen yang digunakan merupakan data nasional. Data historis harga cabai ditunjukkan pada Tabel 3.1.

**Tabel 3.1** Data historis harga cabai

Periode	Harga Cabai (Rp)	Produksi Cabai (ton)	Konsumsi Cabai (ton)	Luas Panen (Ha)
Jan-2010	18.224,00	65.655,220	30.005,338	19.643,880
Feb-2010	19.488,00	65.828,080	30.073,456	19.665,800
Mar-2010	15.179,00	66.036,790	30.134,795	19.687,410
⋮	⋮	⋮	⋮	⋮
Okt-2015	22.958,00	90.349,060	31.698,057	20.641,720
Nop-2015	22.774,00	90.156,990	31.743,500	20.436,350
Des-2015	32.519,00	89.936,710	31.786,600	20.217,100

Sumber: (Badan Pusat Statistik, 2015)

Selengkapnya dapat dilihat pada Lampiran 1.

### 3.2.4 Peralatan Pendukung

Untuk menunjang penelitian peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* Peralatan pendukung dalam penelitian diperlukan beberapa komponen perangkat keras (hardware) dan perangkat lunak (software). Perangkat yang digunakan dalam pembuatan dan penelitian ini adalah sebagai berikut:

#### 3.2.4.1 Perangkat Keras

Perangkat keras yang digunakan dalam mengembangkan sistem ini antara lain:

- Notebook dengan spesifikasi CPU AMD A4-3305M
- GPU AMD Radeon™ HD 6480G
- RAM DDR3 3 GB
- Harddisk 500GB.

### 3.2.4.1 Perangkat Lunak

Perangkat lunak yang digunakan dalam mengembangkan sistem ini antara lain:

- Sistem Operasi Windows 10 Home Premium 64-bit,
- Netbeans IDE 8.0 sebagai editor bahasa pemrograman Java,
- Mozilla Firefox versi 59.0.1 sebagai website browser,
- Microsoft Word 2013 sebagai editor untuk menyusun dokumen penelitian,
- Microsoft Excel 2013 yang digunakan untuk menghitung manualisasi algoritme dalam penelitian ini.

### 3.4 Jadwal Penelitian

Dalam pengerjaan penelitian penulis membutuhkan serangkaian kegiatan yang dilakukan untuk mendukung kegiatan penelitian. Serangkaian kegiatan yang dilakukan tersebut digambarkan dalam bentuk rencana jadwal penelitian. Adapun rencana jadwal penelitian dijelaskan dalam Tabel 3.2.

**Tabel 3.2** Rencana jadwal penelitian

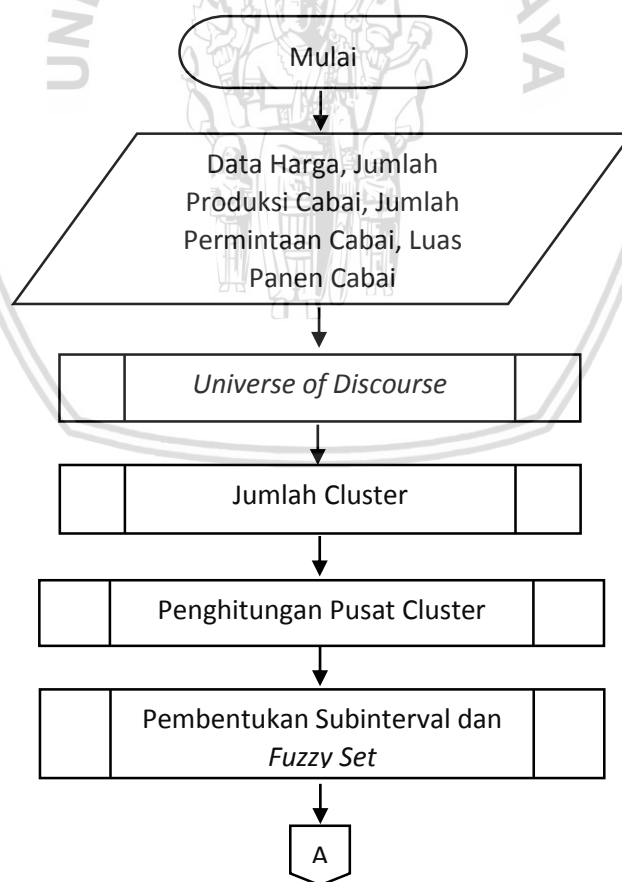
No	Kegiatan	Minggu ke -													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Perancangan penelitian														
2	Implementasi penelitian														
3	Percobaan penelitian														
4	Pengujian penelitian														
5	Analisis penelitian														

## BAB 4 PERANCANGAN

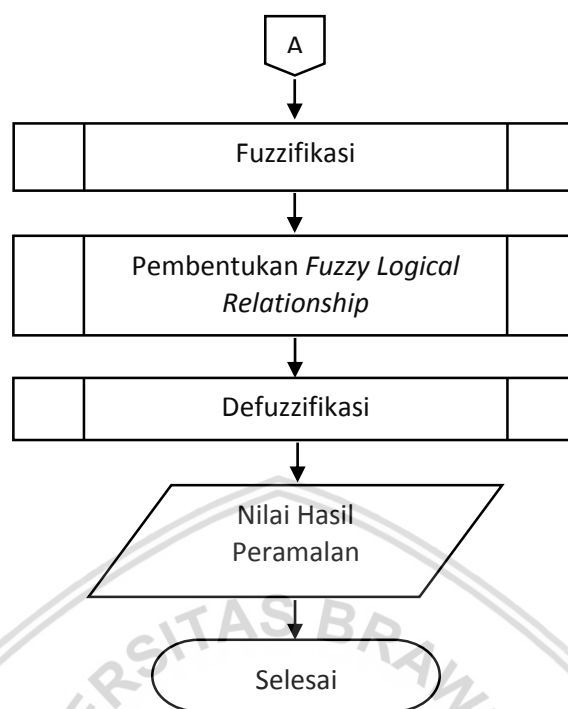
Pada bab ini berisi tentang perancangan yang digunakan sebagai dasar dalam tahap implementasi. Pada bab ini yang akan dibahas meliputi perancangan proses, perhitungan manual, perancangan pengujian, dan perancangan antar muka.

### 4.1 Perancangan Proses

Pada perancangan proses berisi tahapan–tahapan dari metode yang digunakan pada penelitian ini, yaitu metode *High Order Fuzzy Times Series Multifactors*. Penelitian ini melibatkan faktor–faktor yang memengaruhi harga cabai, yaitu harga pada periode sebelumnya, jumlah produksi, jumlah konsumsi dan luas lahan panen. Data time series dari harga cabai merupakan faktor utama dalam proses peramalan pada penelitian ini. *Order* yang digunakan pada metode ini lebih dari satu atau disebut dengan *High Order*. *Order* yaitu urutan jumlah dari data time series pada periode waktu  $t$  yang memengaruhi nilai data. Diagram alir dari proses peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* ditunjukkan pada Gambar 4.1 dan Gambar 4.2.



**Gambar 4.1** Diagram alir metode penelitian yang digunakan (bagian 1)



**Gambar 4.2** Diagram alir metode penelitian yang digunakan (bagian 2)

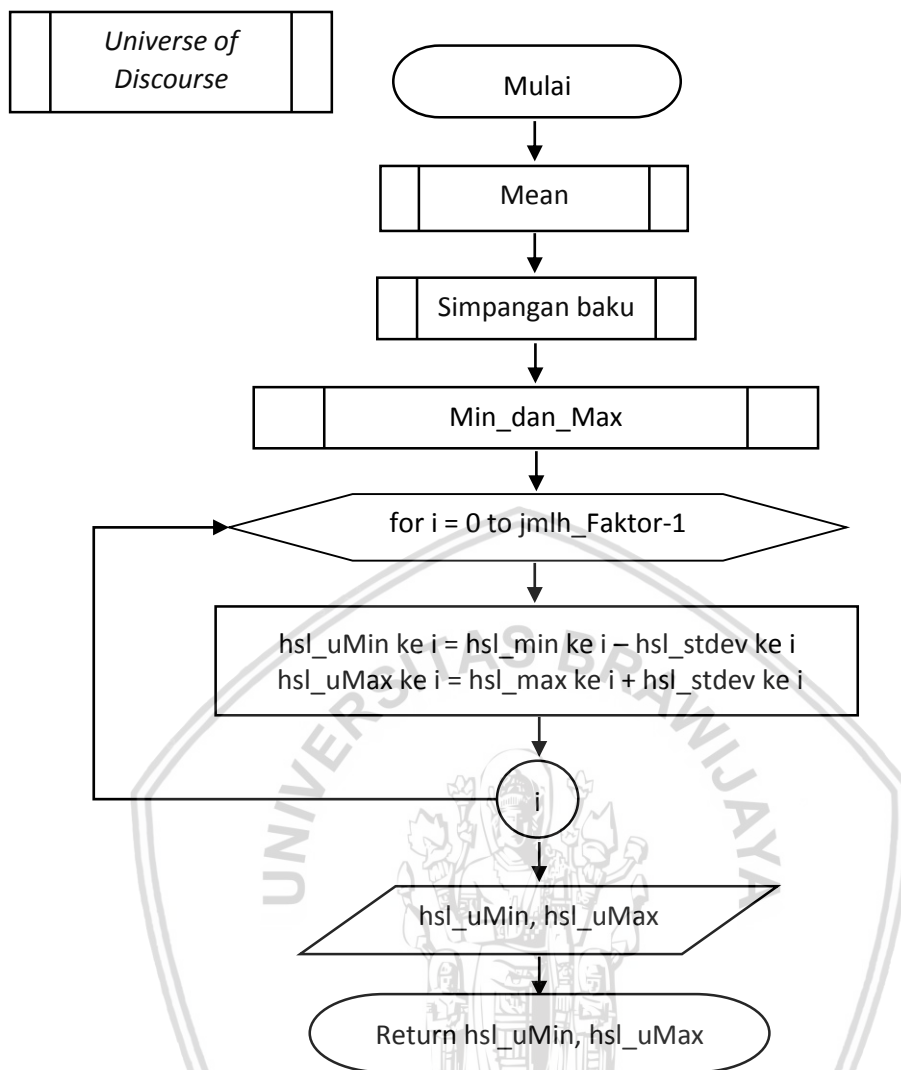
#### 4.1.1 Proses Perhitungan *Universe of Discourse*

Proses perhitungan *Universe of Discourse* digunakan ketika sudah dilakukan inputan data untuk memulai tahapan perhitungan. Tahapan dari proses perhitungan *Universe of Discourse* yaitu:

1. pertama dilakukan perhitungan nilai mean sesuai dengan Persamaan 2.11.
2. Setelah mendapatkan nilai mean dapat dilakukan perhitungan untuk mencari nilai simpangan baku sesuai dengan Persamaan 2.12.
3. Selanjutnya dicari nilai terkecil ( $D_{min}$ ) dan nilai terbesar ( $D_{max}$ ).
4. Setelah menyelesaikan tahap–tahap tersebut dapat dilakukan perhitungan untuk mencari *Universe of Discourse* sesuai dengan Persamaan 2.13.

Berikut merupakan diagram alir dari proses perhitungan *Universe of Discourse* yang ditunjukkan pada Gambar 4.3.

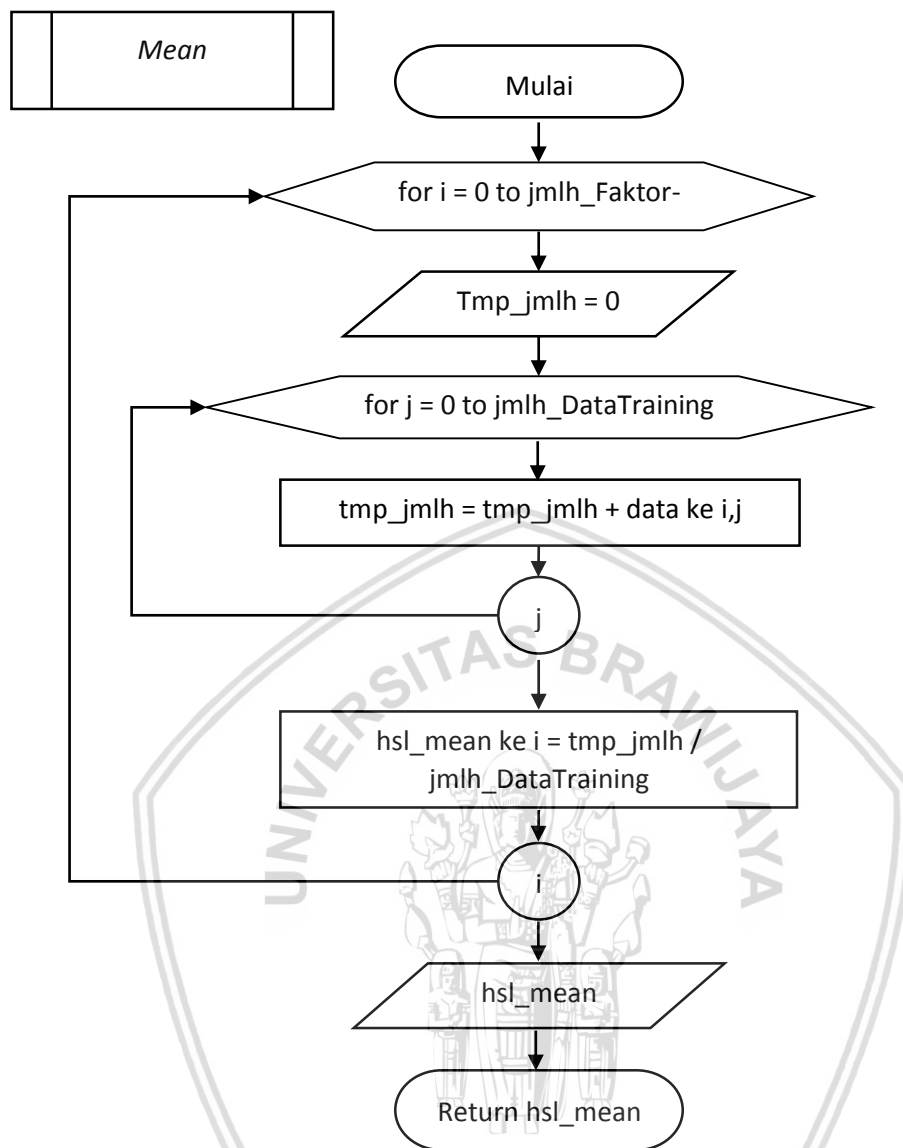




**Gambar 4.3** Diagram alir proses perhitungan *universe of discourse*

#### 4.1.1.1 Proses Perhitungan *Mean*

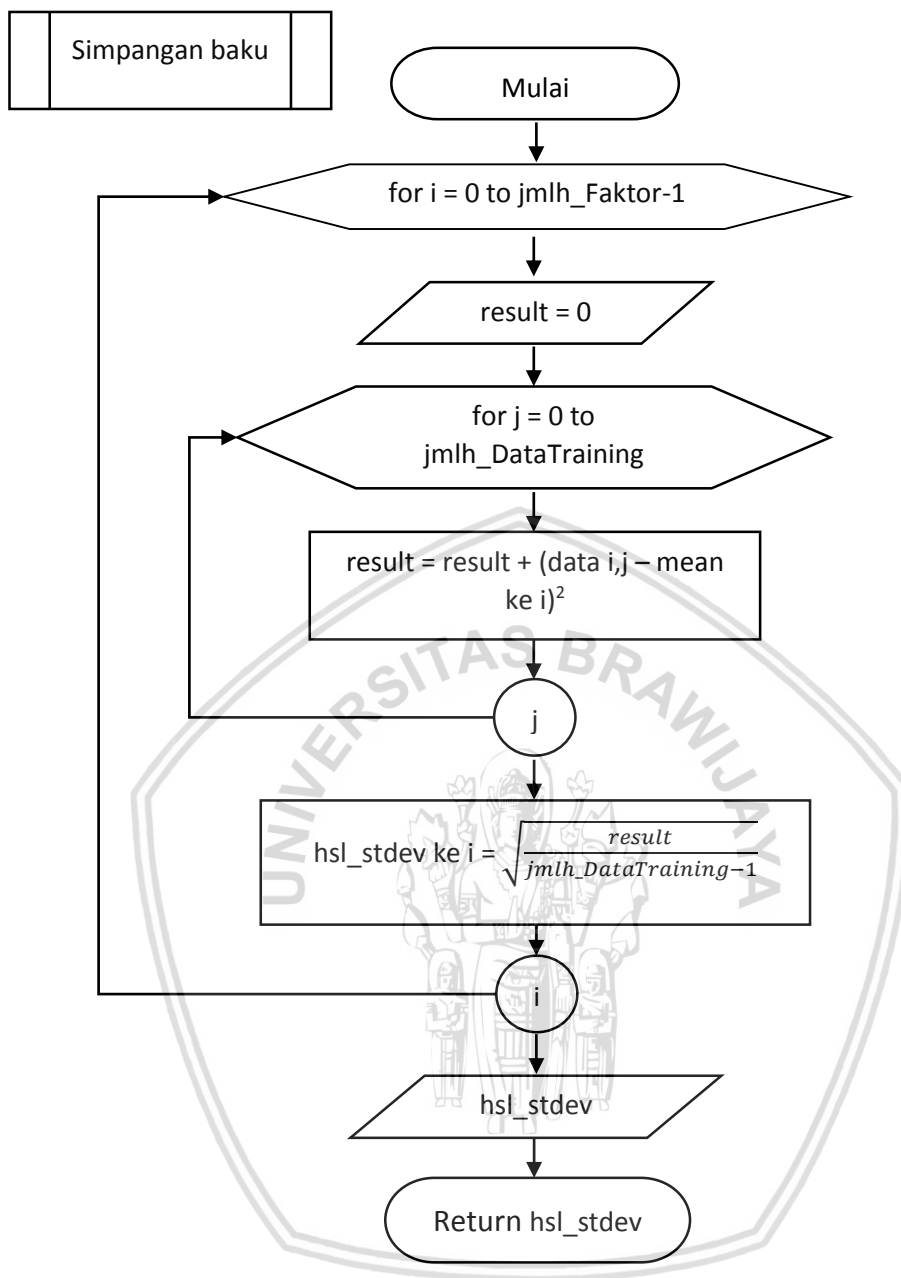
Proses perhitungan *mean* merupakan proses untuk mencari nilai rata-rata dari setiap faktor, dimana nilai *mean* ini dibutuhkan untuk melakukan proses perhitungan simpangan baku. Diagram alir dari proses perhitungan *mean* yang ditunjukkan pada Gambar 4.4.



**Gambar 4.4** Diagram alir proses perhitungan *mean*

#### 4.1.1.2 Proses Perhitungan Simpangan baku

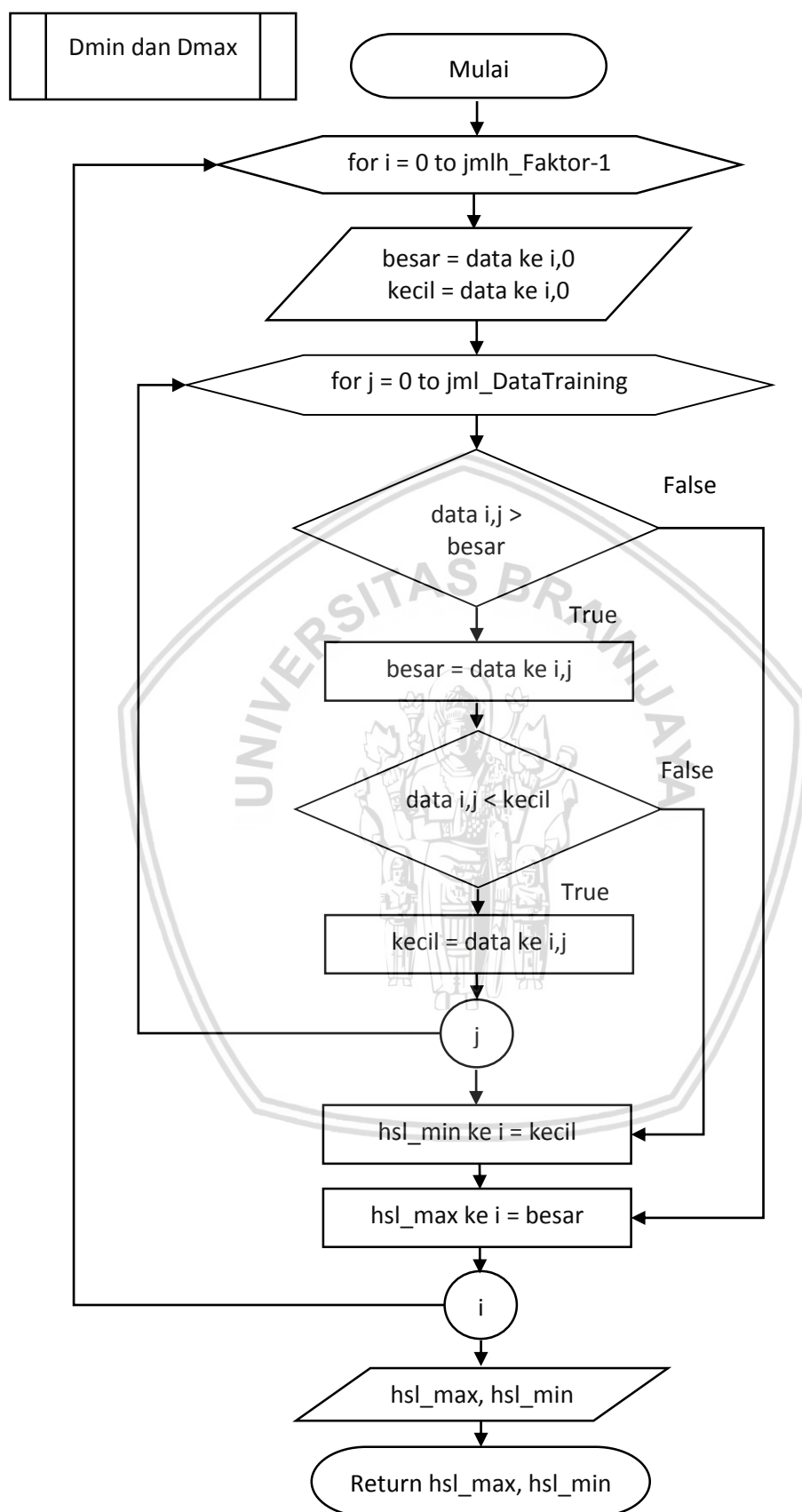
Proses perhitungan simpangan baku dilakukan setelah mendapatkan nilai *mean* dari setiap data sesuai proses sebelumnya. Diagram alir dari proses perhitungan simpangan baku yang ditunjukkan pada Gambar 4.5.



**Gambar 4.5** Diagram alir proses perhitungan simpangan baku

#### 4.1.1.3 Proses Perhitungan Nilai *Dmin* dan *Dmax*

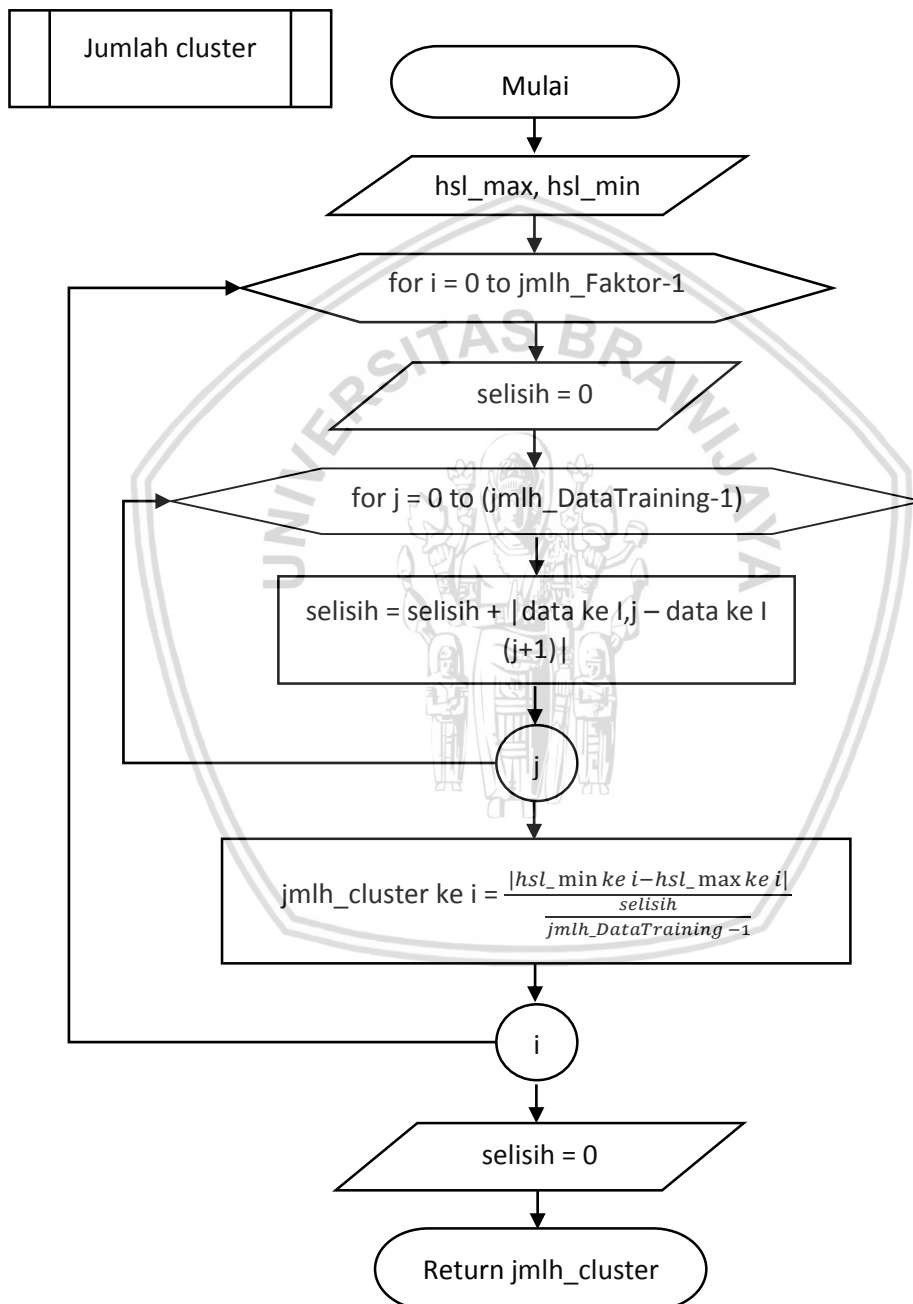
Proses perhitungan nilai *Dmin* dan *Dmax* merupakan proses perhitungan untuk mencari nilai terkecil dan terbesar dari setiap faktor data. Setelah mendapat nilai terkecil dan terbesar maka data dilakukan proses perhitungan *Universe of discourse*. Diagram alir dari proses perhitungan nilai *Dmin* dan *Dmax* yang ditunjukkan pada Gambar 4.6.



**Gambar 4.6** Diagram alir proses perhitungan nilai *Dmin* dan *Dmax*

#### 4.1.2 Proses Perhitungan Jumlah Cluster

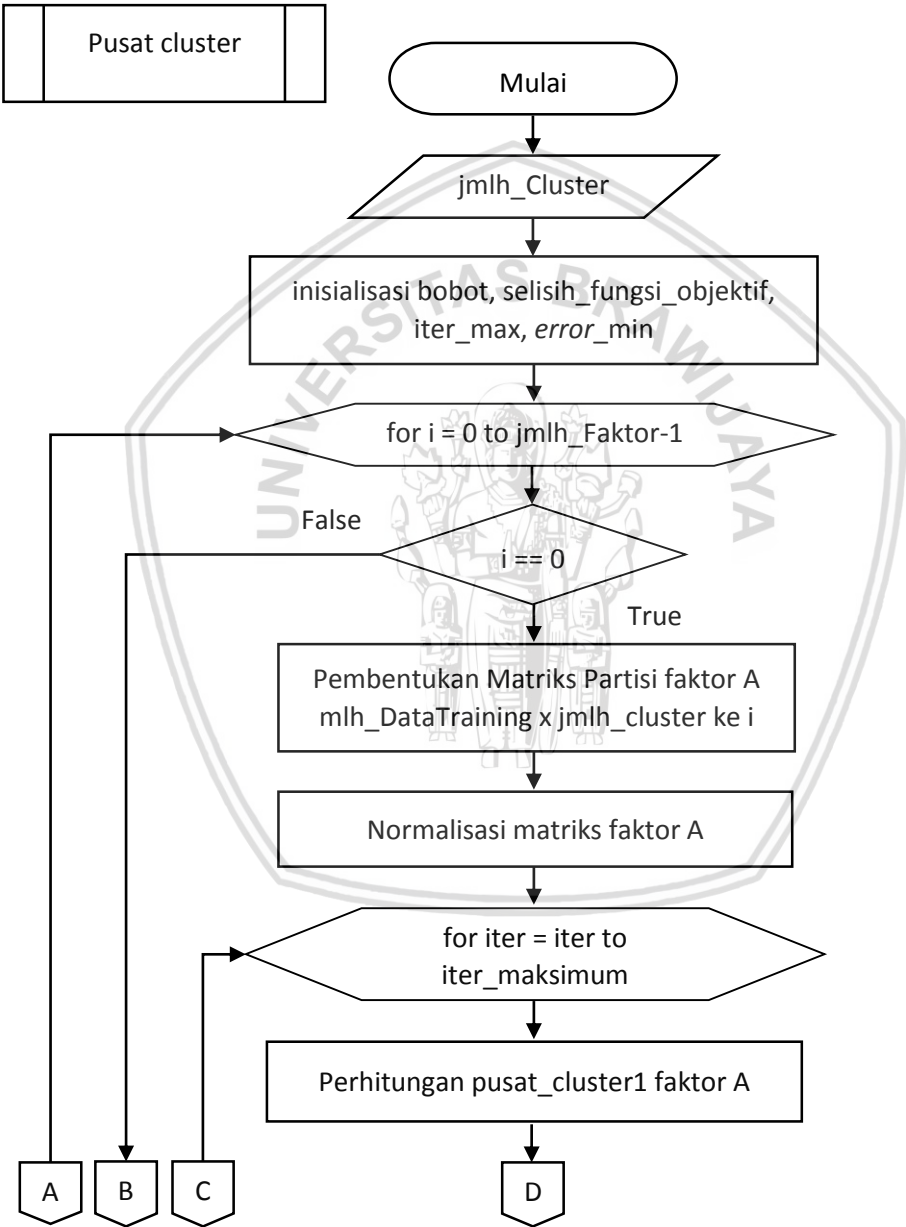
Proses perhitungan jumlah *cluster* merupakan tahapan selanjutnya setelah menentukan *Universe of discourse*. Pada proses ini perhitungan dilakukan untuk mencari nilai dari selisih setiap data. Perhitungan jumlah *cluster* sesuai dengan Persamaan 2.14. Adapun perhitungan jumlah *cluster* ditunjukkan pada Gambar 4.7.



**Gambar 4.7** Diagram alir proses perhitungan jumlah *cluster*

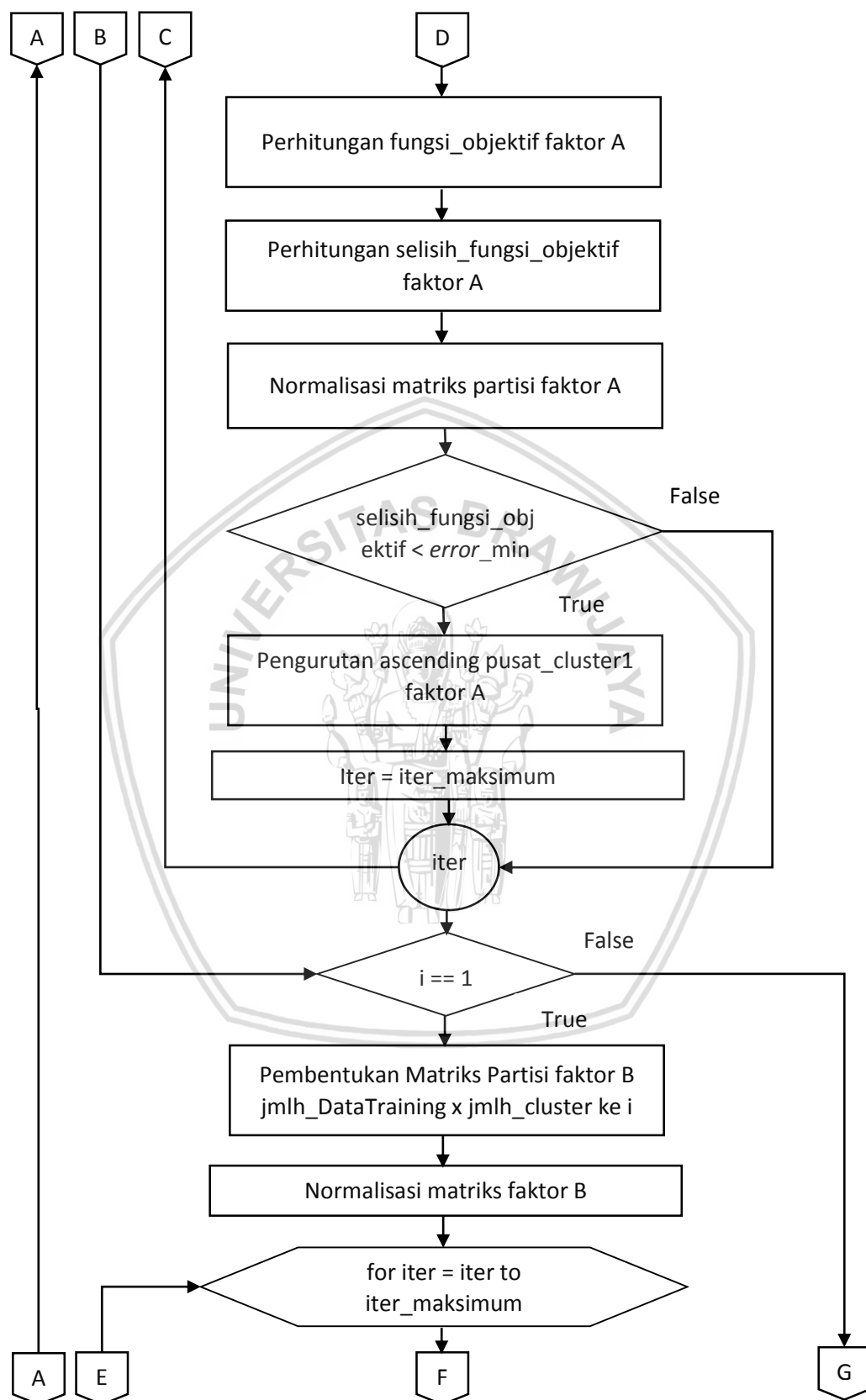
4.1.3 Proses Perhitungan Pusat Cluster

Proses perhitungan pusat *cluster* ini membutuhkan beberapa parameter yang harus dideklarasikan terlebih dahulu yaitu: nilai bobot, fungsi objektif, nilai iterasi awal, nilai iterasi maksimum, dan *error* terkecil. Perhitungan subinterval dalam hal ini menggunakan metode *fuzzy C-means* yang diawali dengan pencarian pusat *cluster* terlebih dahulu. Langkah–langkah perhitungan subinterval yang ditunjukkan pada Gambar 4.8, Gambar 4.9, Gambar 4.10, Gambar 4.11,

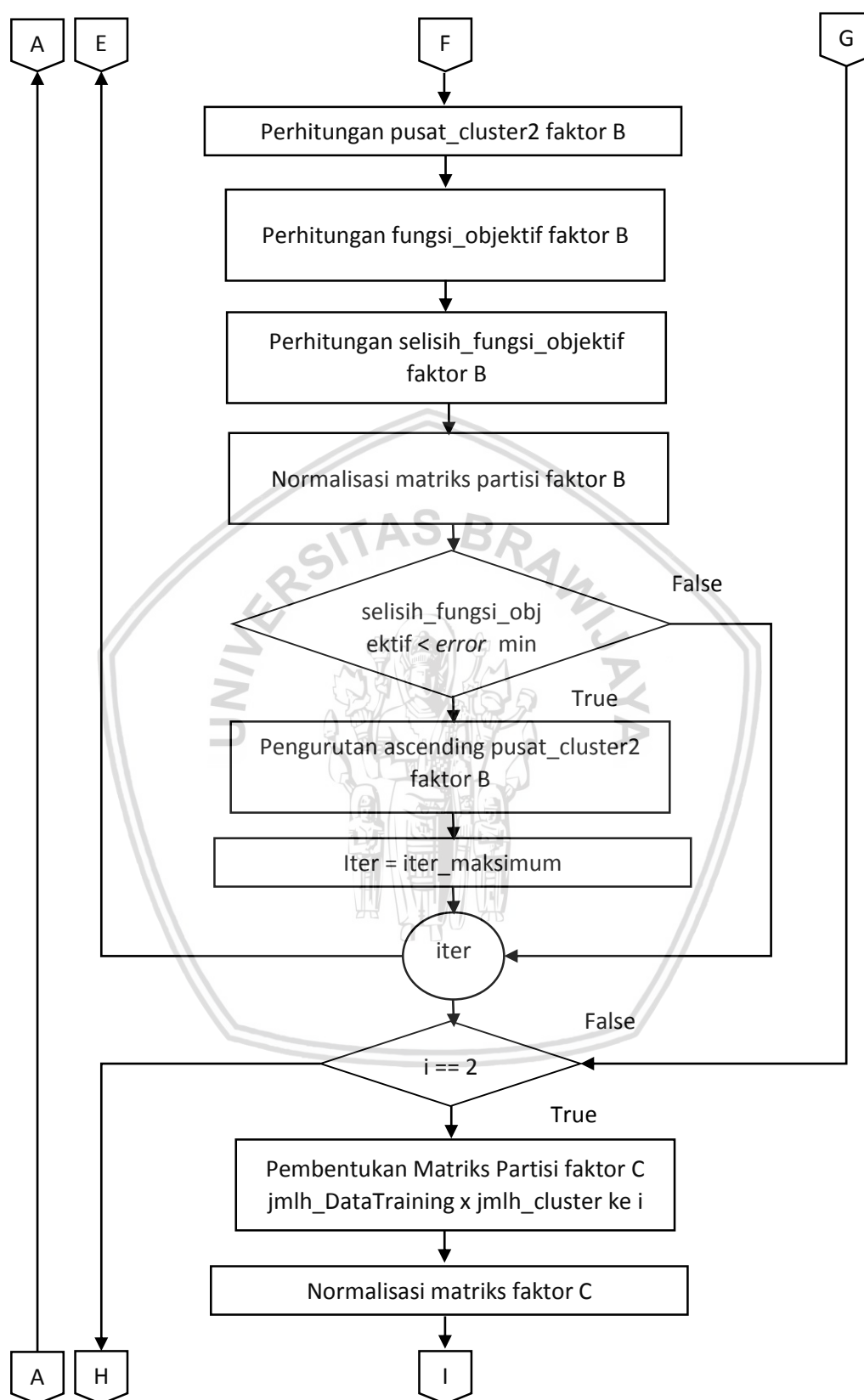


Gambar 4.8 Diagram alir proses perhitungan pusat cluster (bagian 1)

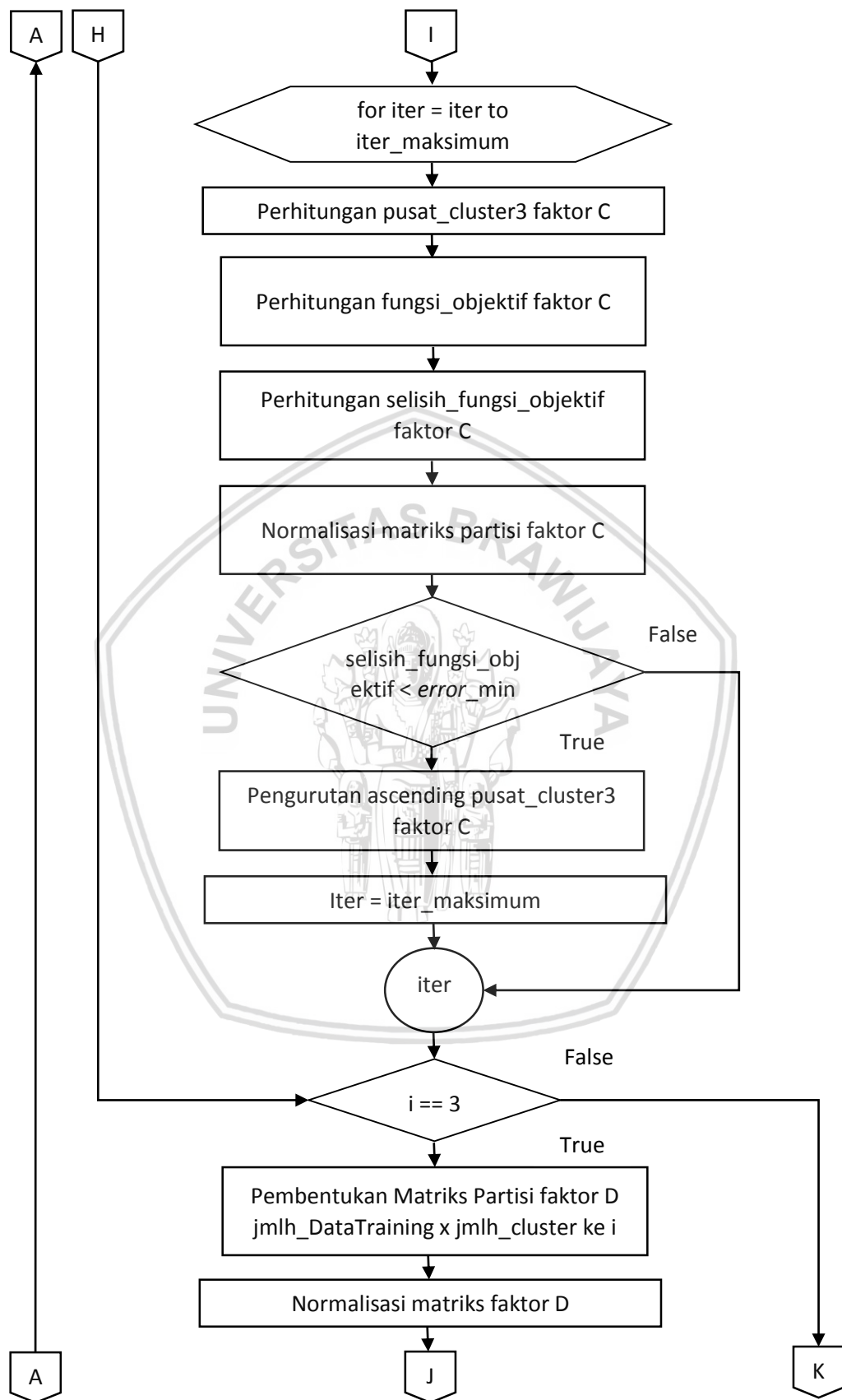




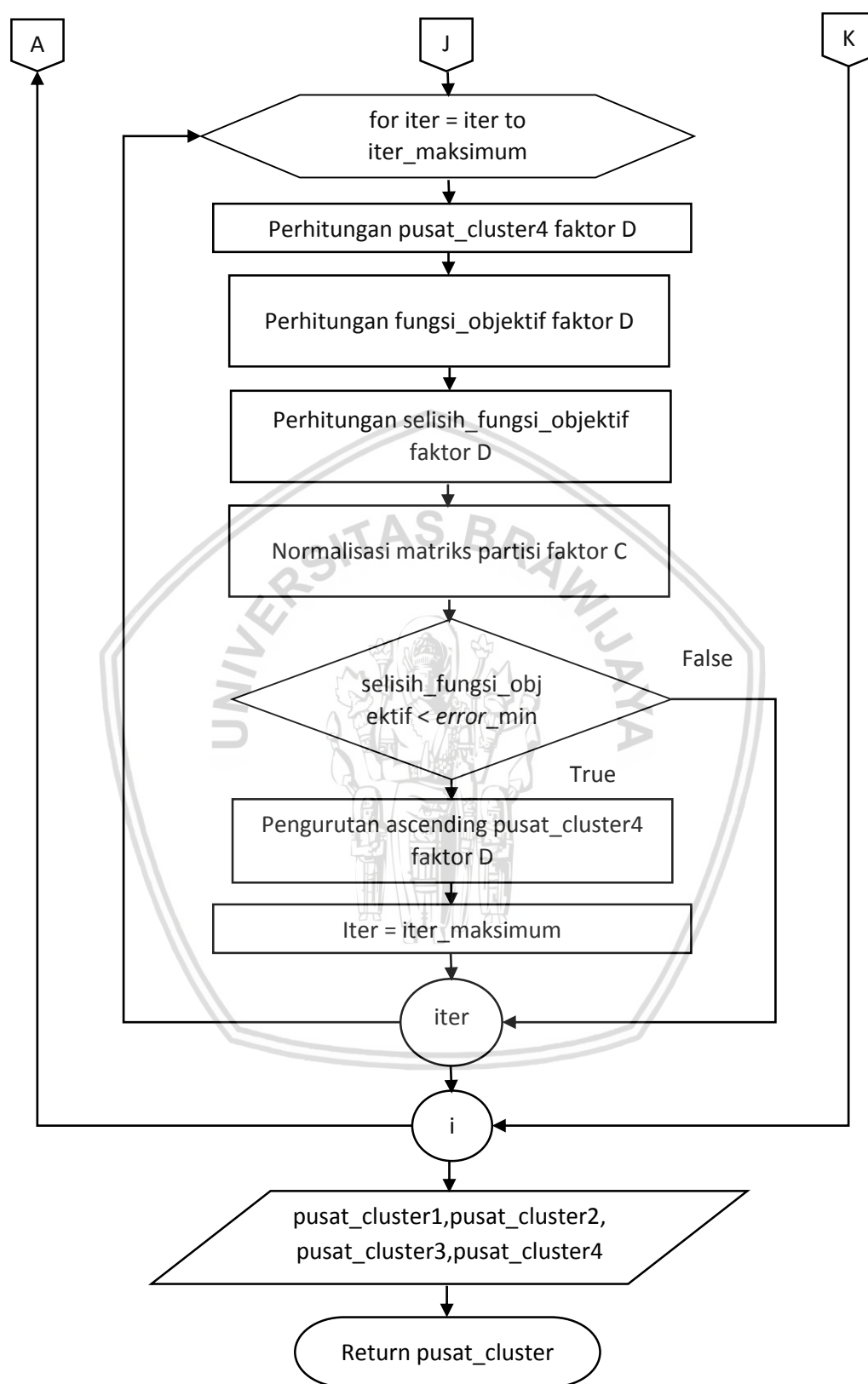
**Gambar 4.9** Diagram alir proses perhitungan pusat *cluster* (bagian 2)



**Gambar 4.10** Diagram alir proses perhitungan pusat *cluster* (bagian 3)



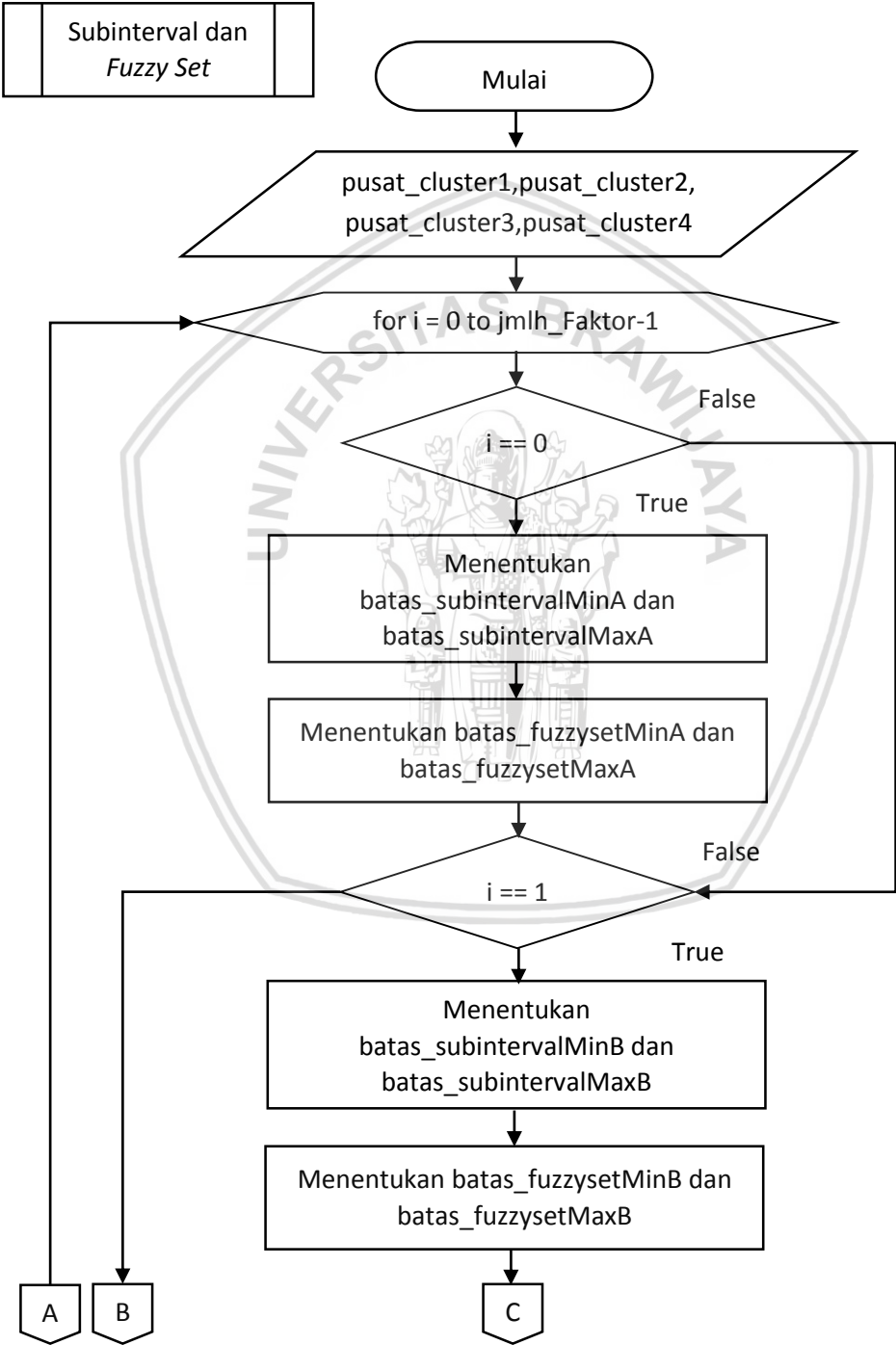
**Gambar 4.11** Diagram alir proses perhitungan pusat *cluster* (bagian 4)



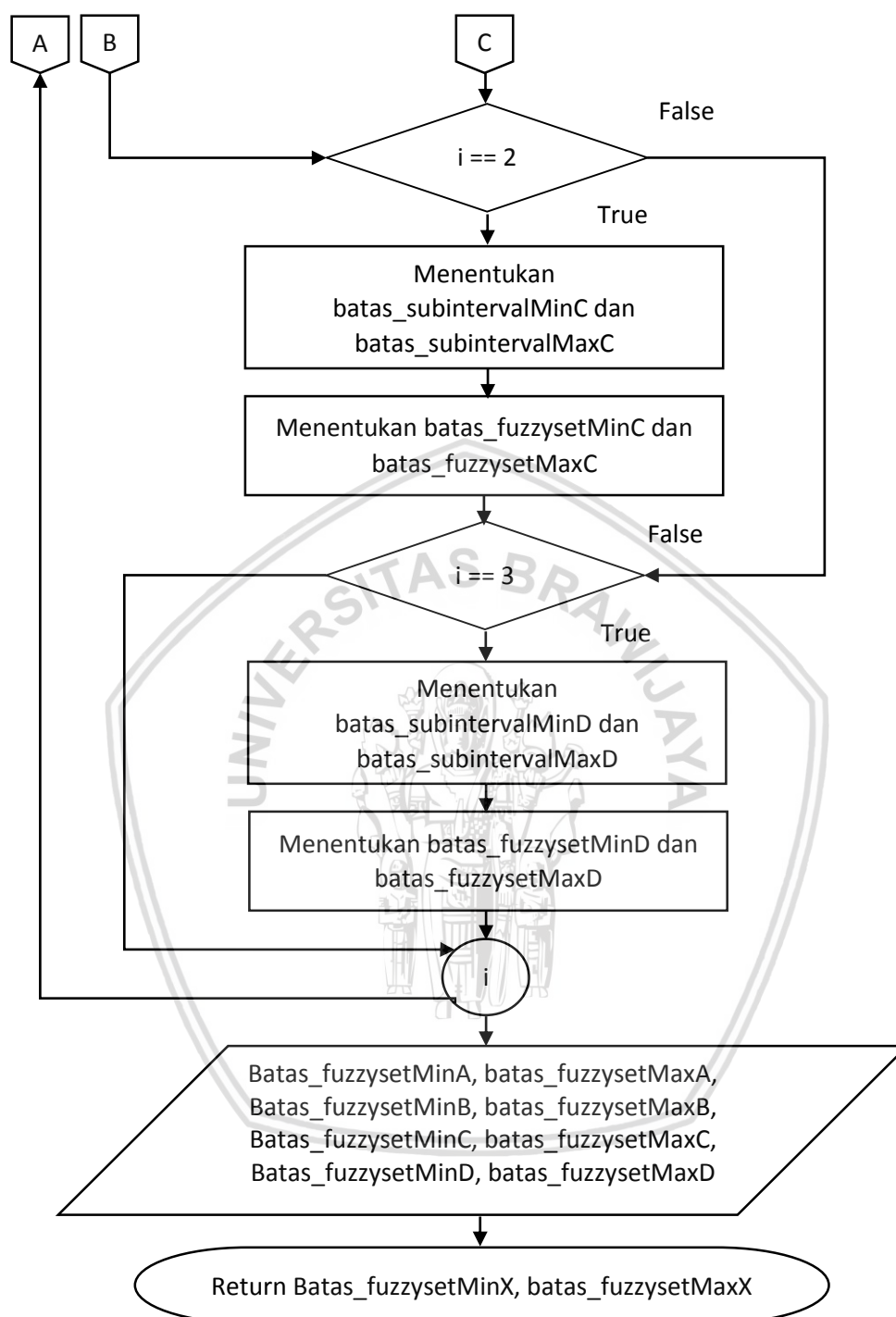
**Gambar 4.12** Diagram alir proses perhitungan pusat *cluster* (bagian 5)

4.1.4 Proses Pembentukan Subinterval dan Fuzzy Set

Proses pembentukan *fuzzy set* atau himpunan fuzzy pada penelitian ini dilakukan setelah perhitungan subinterval selesai dilakukan. Nilai subinterval merupakan inputan utama pada pembentukkan *fuzzy set*. Pembentukan *fuzzy set* sesuai dengan Persamaan 2.15. Langkah–langkah proses pembentukan *fuzzy set* yang ditunjukkan pada Gambar 4.13 dan Gambar 4.14.



Gambar 4.13 Diagram alir pembentukan subinterval dan fuzzy set (bagian 1)

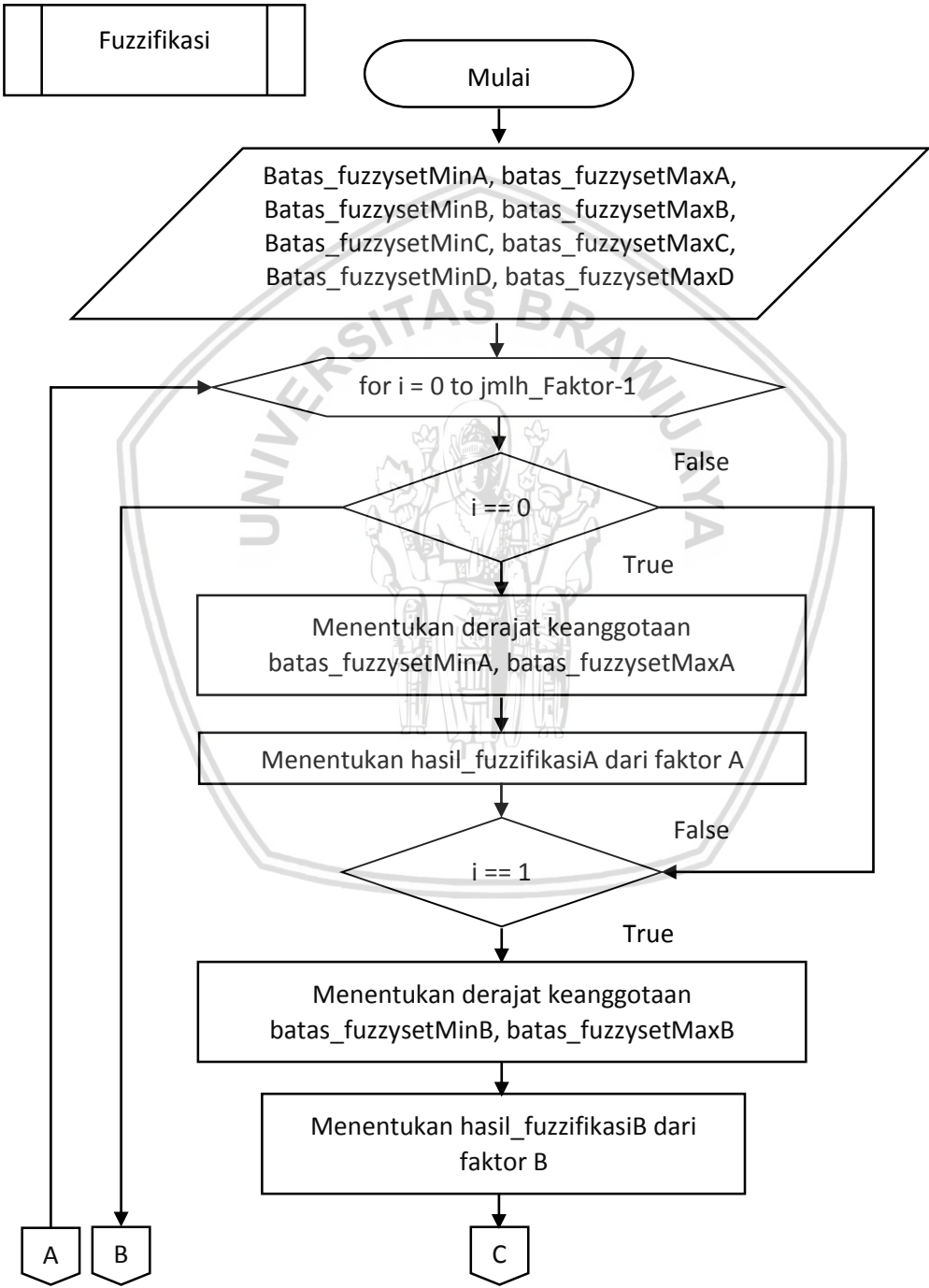


**Gambar 4.14** Diagram alir pembentukan subinterval dan *fuzzy set* (bagian 2)

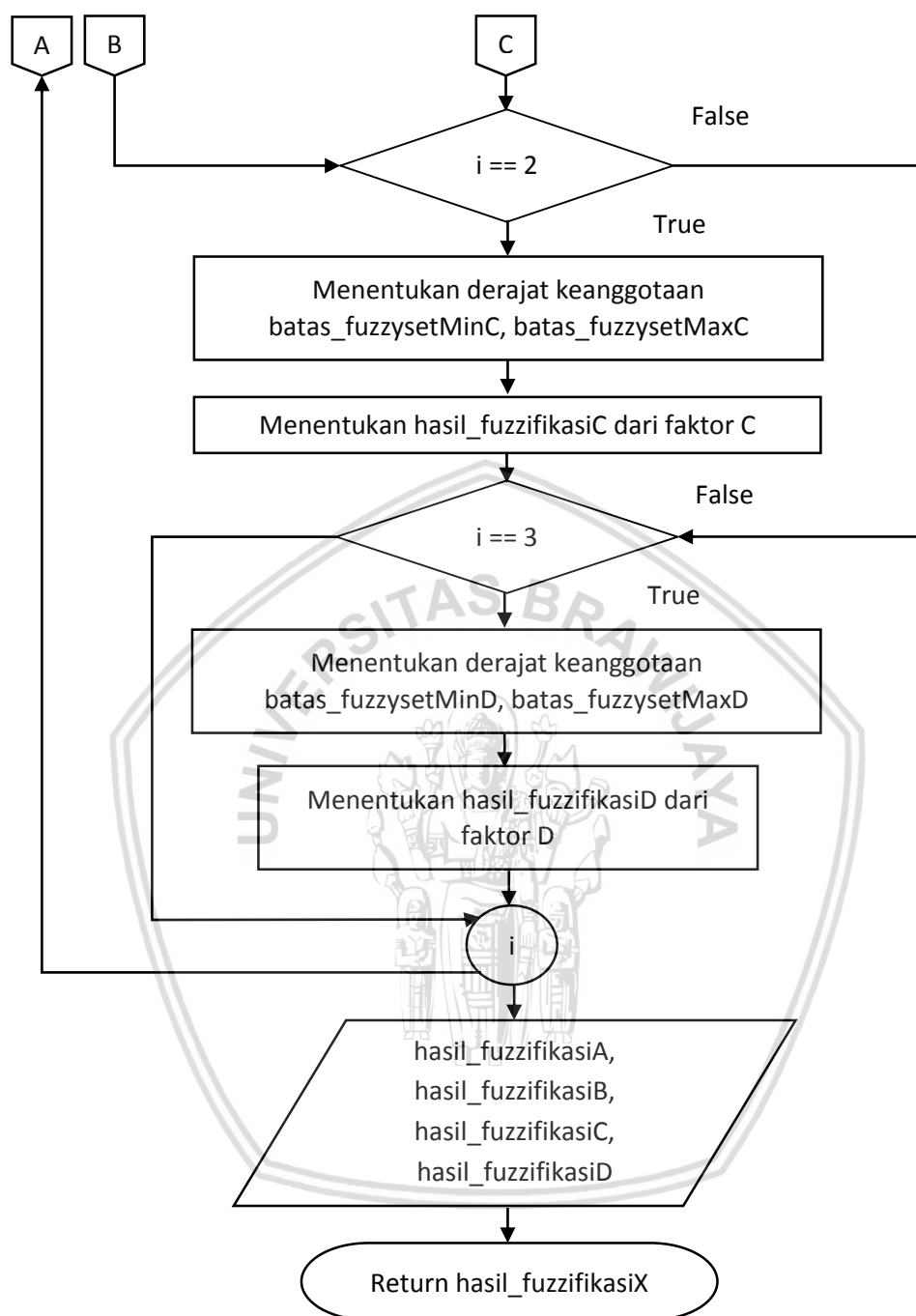


4.1.5 Proses Fuzzifikasi

Proses fuzzifikasi ini pembentukanya sesuai dengan nilai derajat keanggotaan pada setiap data. Pada proses ini, derajat keanggotaan dihitung berdasarkan fungsi keanggotaan kurva bahu. Hasil dari proses ini merupakan data yang sudah difuzzifikasikan menjadi data *fuzzy*. Langkah–langkah dari proses fuzzifikasi ditunjukkan pada Gambar 4.15 dan Gambar 4.16.



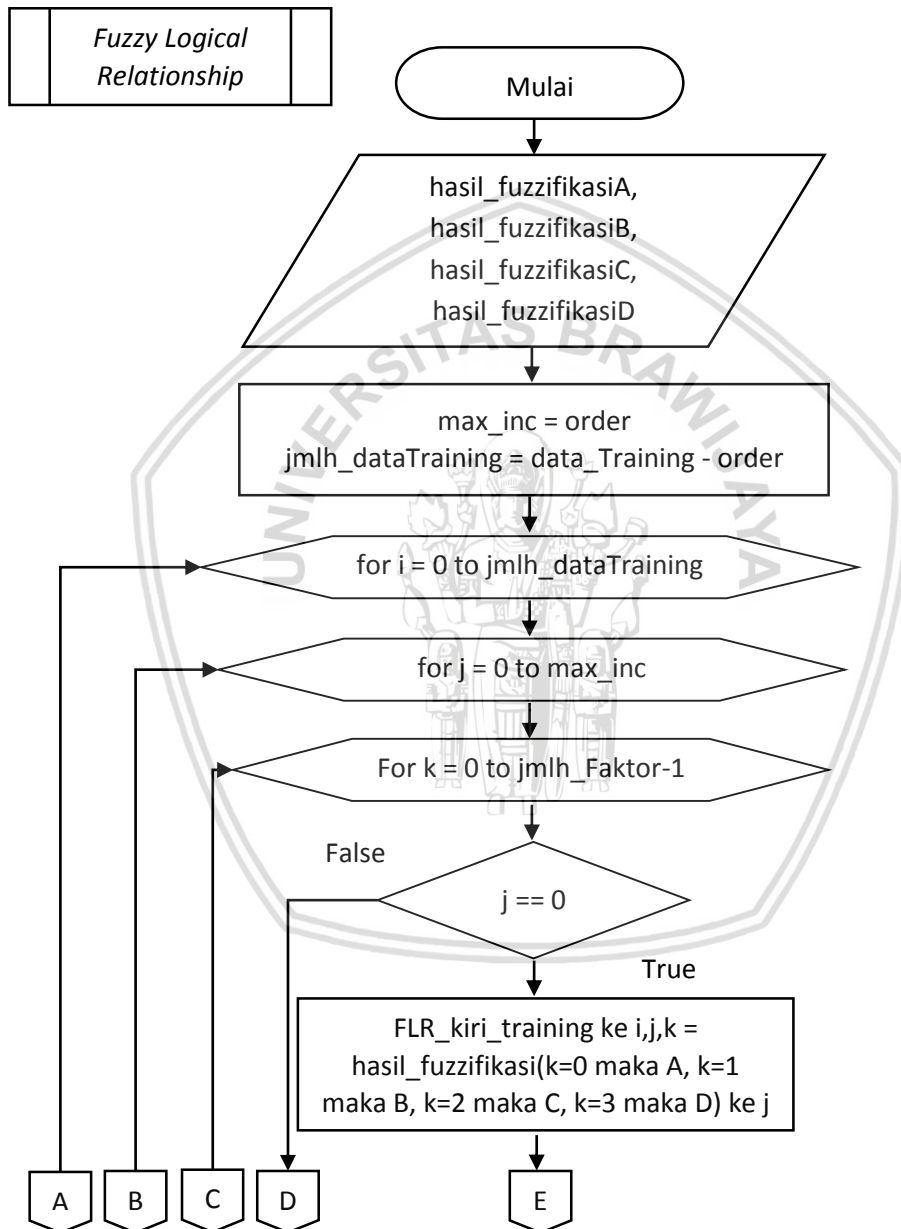
Gambar 4.15 Diagram alir proses fuzzifikasi (bagian 1)



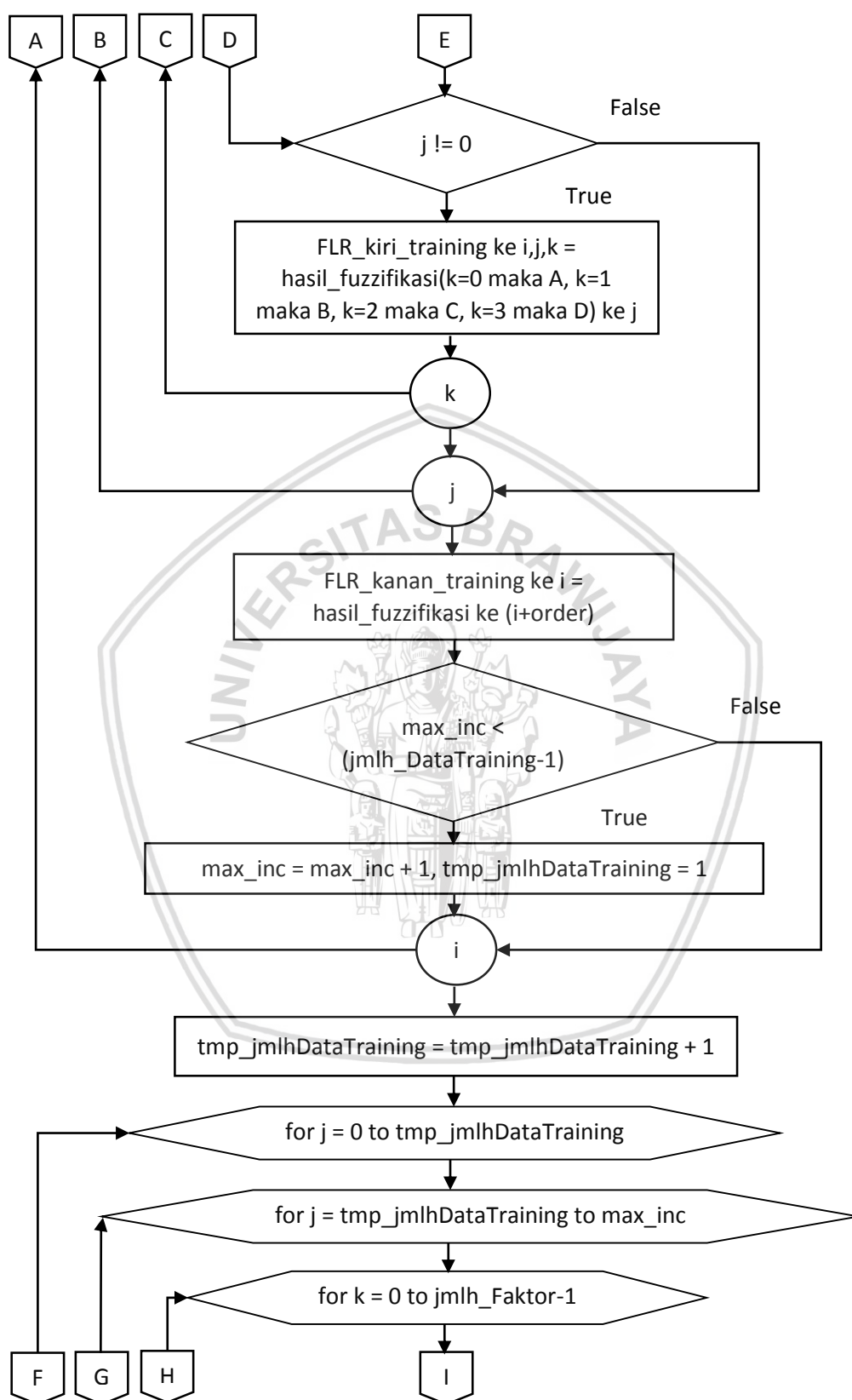
**Gambar 4.16** Diagram alir proses fuzzifikasi (bagian 2)

#### 4.1.6 Proses Pembentukan *Fuzzy Logical Relationship*

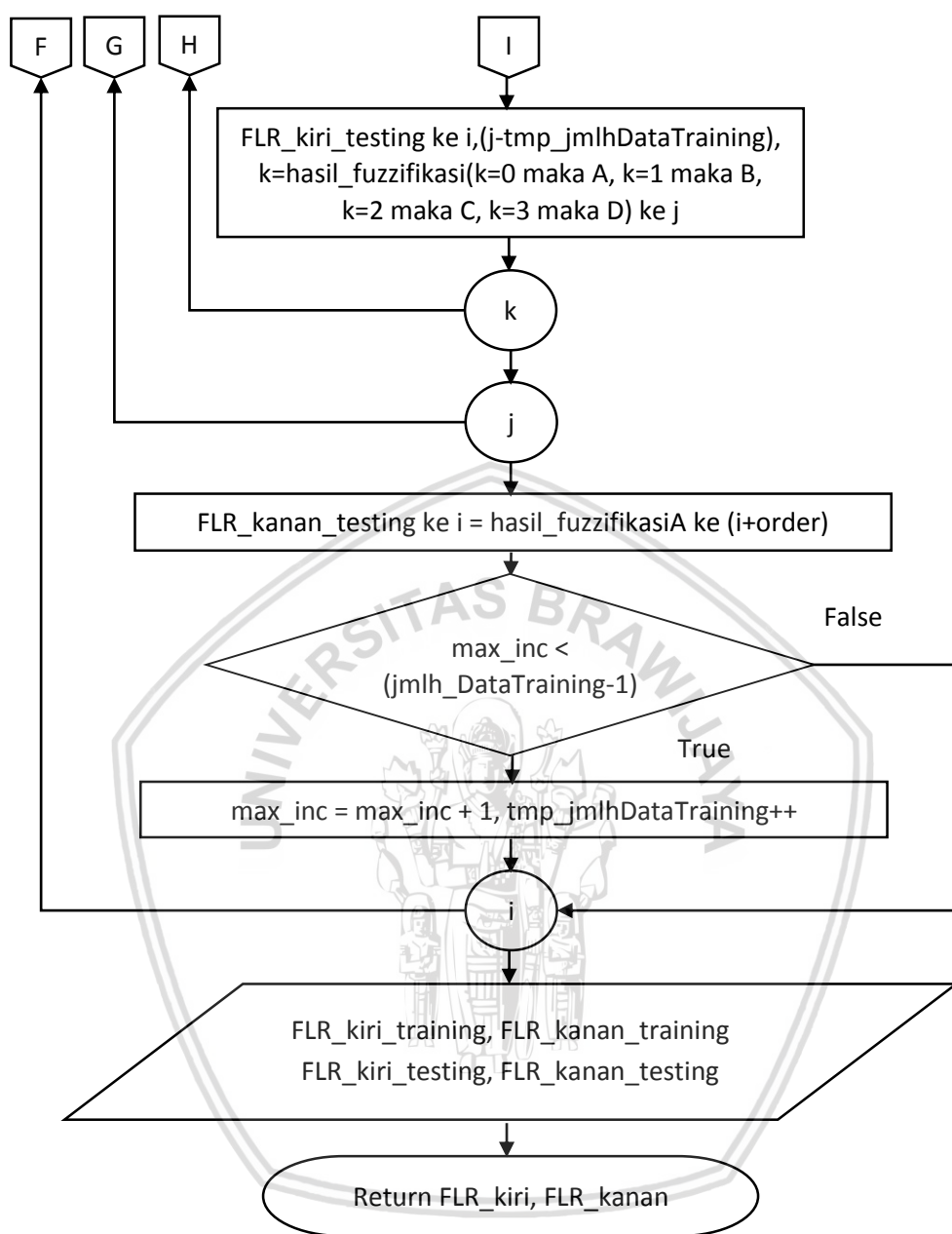
Setelah data melalui proses fuzzifikasi dan sudah dalam bentuk *fuzzy* maka dapat dilakukan pembentukan *Fuzzy Logical Relationship* (FLR). Pembentukan FLR dilakukan sesuai dengan Persamaan 2.16. Langkah–langkah proses pembentukan *Fuzzy Logical Relationship* (FLR) yang ditunjukkan pada Gambar 4.17, Gambar 4.18, dan Gambar 4.19.



**Gambar 4.17** Diagram alir proses pembentukan FLR (bagian 1)



**Gambar 4.18** Diagram alir proses pembentukan FLR (bagian 2)



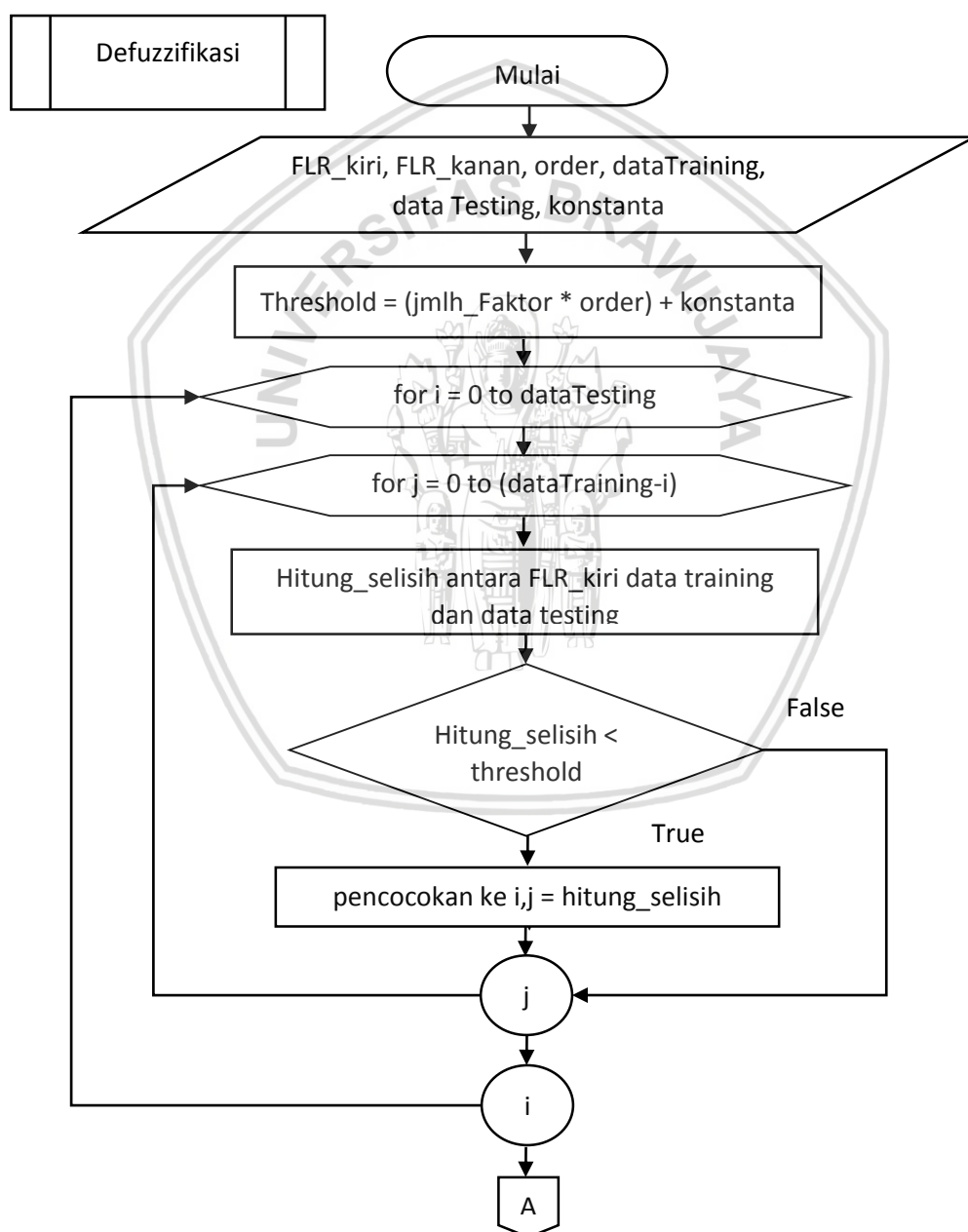
**Gambar 4.19** Diagram alir proses pembentukan FLR (bagian 3)

#### 4.1.7 Proses Defuzzifikasi

Proses defuzzifikasi merupakan tahap akhir dimana *output* dari proses ini merupakan nilai hasil peramalan. Data yang diperlukan pada proses defuzzifikasi adalah hasil dari proses pembentukan *Fuzzy Logical Relationship* (FLR), nilai *threshold* yang harus dideklarasikan sebelumnya, jumlah data *training*, dan jumlah data *testing*. Dalam proses ini dilakukan perhitungan selisih pada FLR antara data *testing* dan data *training*. FLR yang dihitung nilai selisihnya adalah FLR pangkat bawah yang terdapat pada ruas kiri. Setelah dilakukan perhitungan selisih dan

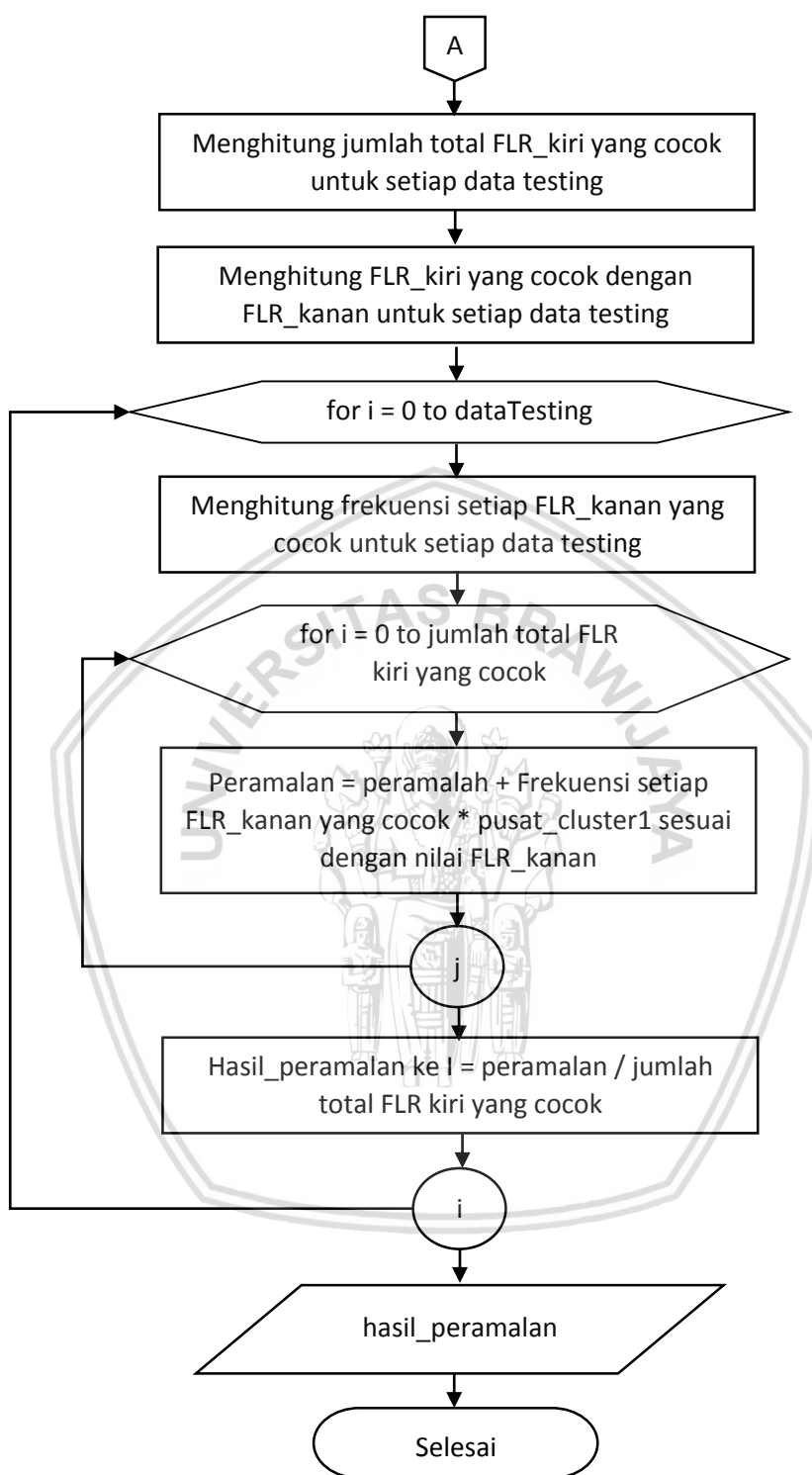
total jumlah selisih lebih kecil dari nilai *threshold*, maka pada setiap FLR yang sesuai dengan kondisi tersebut dapat dikatakan cocok atau dapat dilakukan ke tahap perhitungan selanjutnya.

Setelah mendapatkan FLR yang cocok maka dilakukan pengelompokkan berdasarkan FLR ruas kanan. Pengelompokkan ini dihitung jumlah frekuensi yang sama sesuai dengan *fuzzy set* di ruas kanan. Selanjutnya untuk memperoleh hasil peramalan dilakukan proses perhitungan sesuai dengan Persamaan 2.18. Langkah-langkah proses defuzzifikasi yang ditunjukkan pada Gambar 4.20 dan Gambar 4.21.



**Gambar 4.20** Diagram alir proses defuzzifikasi (bagian 1)





**Gambar 4.21** Diagram alir proses defuzzifikasi (bagian 2)

## 4.2 Perhitungan Manual

Dalam perhitungan manual ini data sampel yang digunakan merupakan data *time series* harga cabai tingkat nasional periode Januari 2010 sampai dengan Desember 2015. Namun dalam tahap perhitungan manual ini hanya digunakan data sampel sebanyak 10 data dengan rentang sampel antara Januari 2012 sampai Oktober 2012. Adapun data sampel ditunjukkan pada Tabel 4.1.

**Tabel 4.1** Data sampel perhitungan manual

No.	Periode	Harga (A)	Produksi Cabai (B)	Konsumsi Cabai (C)	Luas Panen (D)
1	Januari 2012	27533,00	77120,720	33453,105	20039,120
2	Februari 2012	20697,00	77571,300	33700,382	20059,620
3	Maret 2012	21588,00	78017,900	33892,686	20082,590
4	April 2012	23147,00	78460,510	34030,016	20108,030
5	Mei 2012	22240,00	78899,130	34112,372	20135,940
6	Juni 2012	27086,00	79333,760	34139,754	20166,330
7	Juli 2012	26118,00	79764,410	34112,161	20199,180
8	Agustus 2012	23844,00	80191,080	34029,595	20234,500
9	September 2012	19441,00	80613,750	33892,055	20272,300
10	Januari 2012	20565,00	81032,440	33699,542	20312,570

Sumber: (Badan Pusat Statistik, 2015)

### 4.2.1 Perhitungan *Universe of Discourse*

Untuk melakukan perhitungan *Universe of Discourse*, harus dicari terlebih dahulu nilai  $D_{min}$ ,  $D_{max}$ , dan simpangan baku.  $D_{min}$  adalah nilai minimum atau nilai terkecil dari sampel data yang ada, sedangkan  $D_{max}$  adalah nilai maksimum atau nilai paling besar dari sampel data yang ada. Untuk mencari nilai simpangan baku harus menghitung nilai *mean* (rata-rata) terlebih dahulu sesuai dengan Persamaan 2.17. *Mean* dari data harga cabai (A) dihitung dengan cara sebagai berikut.

$$\text{Mean} = (\text{data ke-1} + \text{data ke-2} + \text{data ke-3} + \dots + \text{data ke-10}) / 10$$

$$= (27533,00 + 20697,00 + 21588,00 + 23147,00 + 22240,00 + 27086,00 + 26118,00 + 23844,00 + 19441,00 + 20565,00) / 10$$

$$= 232264,00 / 10$$

$$= 23226,418$$

Setelah mendapat nilai *mean* (rata-rata), dilanjutkan dengan melakukan perhitungan simpangan baku sesuai dengan Persamaan 2.18. Simpangan baku adalah akar kuadrat dari *varians*. Simpangan baku juga merupakan ukuran keragaman data statistik atau bagaimana bentuk sebaran data dalam sampel. Untuk menghitung simpangan baku agar lebih mudah kita terlebih dahulu melakukan perhitungan nilai kuadrat dari selisih data ke-*i* dan *mean* yang ditunjukkan pada Tabel 4.2.

**Tabel 4.2** Hasil kuadrat dari selisih data ke-*i* dan *mean*

i	Data ke-i	Data ke-i - <i>mean</i>	(Data ke-i - <i>mean</i> ) <sup>2</sup>
1	27533,00	4307,372	18553453,546384
2	20697,00	-2528,838	6395021,630244
3	21588,00	-1637,868	2682611,585424
4	23147,00	-79,288	6286,586944
5	22240,00	-986,408	973000,742464
6	27086,00	3860,512	14903552,902144
7	26118,00	2891,692	8361882,622864
8	23844,00	618,312	382309,729344
9	19441,00	-3784,578	14323030,638084
10	20565,00	-2660,908	7080431,384464
Total			73661581,368360

Setelah mendapatkan hasil nilai kuadrat dari selisih data ke-*i* dan *mean* sesuai dengan Tabel 4.2, dilanjutkan dengan melakukan perhitungan simpangan baku dengan cara sebagai berikut.

$$\begin{aligned}
 \text{Simpangan\_baku} &= \sqrt{\frac{1}{(10-1)} \sum_{i=1}^{10} (\text{data ke} - i - \text{mean})^2} \\
 &= \sqrt{\frac{1}{9}} \times 73661581,368360 \\
 &= 2860,877514
 \end{aligned}$$

Untuk perhitungan simpangan baku juga dilakukan pada faktor lainnya yaitu produksi cabai (B), konsumsi cabai (C), dan luas panen (D). Hasil perhitungan *Dmin*, *Dmax*, dan simpangan baku ditunjukkan pada Tabel 4.3.

**Tabel 4.3** Nilai minimum, nilai maksimum dan simpangan baku

Data	<i>Dmin</i>	<i>Dmax</i>	Simpangan Baku
A	19441,840	27533,790	2860,877514
B	77120,720	81032,440	1316,012939
C	33453,105	34139,754	226,268616
D	20039,120	20312,570	92,474690

Berdasarkan hasil yang ditunjukkan Tabel 4.3, selanjutnya dapat dilakukan perhitungan nilai U sesuai dengan Persamaan 2.13. Berikut adalah perhitungan U pada data A.

$$\begin{aligned}
 U &= [Dmin - \sigma, Dmax + \sigma] \\
 &= [(19441,840 - 2860,877514), (27533,790 - 2860,877514)] \\
 &= [16580,962486 \text{ } 30394,667514]
 \end{aligned}$$

Untuk perhitungan U juga dilakukan pada semua faktor yaitu data B, C, dan D. Sehingga diperoleh nilai U yang ditunjukkan pada Tabel 4.4.

**Tabel 4.4** Hasil perhitungan *universe of discourse*

Data	U	
	Batas Bawah	Batas Atas
A	16580,962486	30394,667514
B	75804,707061	82348,452939
C	33226,836384	34366,022616
D	19946,645310	20405,044690

#### 4.2.2 Perhitungan Jumlah *Cluster* (k)

Pada perhitungan jumlah *cluster*, langkah pertama adalah mencari nilai absolut dari selisih data hari ke-t dengan hari sebelumnya. Perhitungan selisih dilakukan hingga data ke-10. Perhitungan nilai selisih dilakukan pada semua data A, B, C, dan D. Selanjutnya masing-masing nilai selisih dijumlah sehingga hasilnya seperti ditunjukkan pada Tabel 4.5.

**Tabel 4.5** Tabel nilai absolut selisih data dengan periode sebelumnya

No.	Periode	A	B	C	D
1	Januari 2012	6836,210	450,580	247,277	20,500
2	Februari 2012	890,970	446,600	192,304	22,970
3	Maret 2012	1558,580	442,610	137,330	25,440
4	April 2012	907,120	438,620	82,356	27,910
5	Mei 2012	4846,920	434,630	27,382	30,390
6	Juni 2012	968,820	430,650	27,593	32,850
7	Juli 2012	2273,380	426,670	82,566	35,320
8	Agustus 2012	4402,890	422,670	137,540	37,800
9	September 2012	1123,670	418,690	192,513	40,270
Jumlah		23808,560	3911,720	1126,861	273,450

Setelah mendapatkan nilai absolut selisih data hari ke-t dengan hari sebelumnya, dapat dilanjutkan dengan menghitung jumlah *cluster* sesuai dengan Persamaan 2.20. Perhitungan jumlah *cluster* pada data A (k1) berdasarkan pada Tabel 4.3 diketahui nilai  $D_{min}$  adalah 19441.840, nilai  $D_{max}$  adalah 27533.790, maka jumlah *cluster* data A dapat ditunjukkan sebagai berikut.

$$k1 = \frac{|D_{min} - D_{max}|}{\frac{\sum_{t=1}^n |X(t) - X(t-1)|}{n-1}}$$

$$k1 = 19441,840 - \frac{27533,790}{\frac{24262,360}{10-1}} =$$

$$k1 = -3$$

Apabila hasil perhitungan jumlah *cluster* merupakan bilangan pecahan maka hasilnya harus dibulatkan. Sehingga jumlah *cluster* pada data A adalah 3. Perhitungan jumlah *cluster* juga dilakukan pada data B (k2), C (k3), dan D (k4). Berikut merupakan hasil semua perhitungan jumlah *cluster* yang ditunjukkan pada Tabel 4.6.

**Tabel 4.6** Hasil jumlah *cluster* data

No.	Cluster	Jumlah Cluster
1	K1	3
2	K2	9
3	K3	5
4	K4	9

### 4.2.3 Perhitungan Pusat *Cluster* dan Subinterval

Pada proses pembentukan subinterval Tabel 4.6 menunjukkan jumlah *cluster* yang selanjutnya digunakan untuk menentukan pusat *cluster* dan subinterval menggunakan algoritma *fuzzy C-means* (FCM).

#### 4.2.3.1 Penentuan Pusat *Cluster*

##### a. Penentuan Nilai Variabel

Pada penentuan subinterval diawali dengan klasterisasi menggunakan algoritma *fuzzy C-means* (FCM) pada awal proses ditentukan beberapa variabel sebagai berikut:

- Banyak *cluster* =  $k_1 = 2$
- Bobot ( $w$ ) = 2
- Fungsi objektif awal = 0
- Iterasi awal = 1
- Iterasi maksimum = 100
- *Error* terkecil = 0,01

##### b. Pembentukan Matriks Partisi Awal ( $\mu$ )

Data pada pembentukan matriks partisi awal merupakan nilai random antara 0 sampai 1. Dimensi pada matriks partisi awal dibentuk dengan dimensi jumlah data ( $i$ ) x jumlah *cluster* ( $k$ ). Dalam hal ini telah diketahui jumlah data adalah 10 dan  $k_1$  adalah 2, maka partisi dimensi matriks awal adalah 10 x 2. Hasil pembentukan matriks partisi awal ditunjukkan pada Tabel 4.7 dimana *Cluster* disimbolkan dengan C.



**Tabel 4.7** Matriks partisi awal pada *cluster* data A

i	$\mu$	
	C1	C2
1	0,354509	0,170157
2	0,173770	0,815438
3	0,200405	0,190224
4	0,386712	0,455563
5	0,597699	0,526087
6	0,818713	0,170134
7	0,701820	0,231532
8	0,719487	0,572203
9	0,349173	0,101522
10	0,851763	0,321522

c. Normalisasi Matriks Partisi Awal

Langkah pertama proses normalisasi matriks partisi adalah dengan menghitung jumlah bilangan penyusun matriks pada setiap kolom yang dapat dilihat pada Tabel 4.7 dijumlah berdasarkan dengan Persamaan 2.2. Diketahui jumlah kolom matriks sebanyak 2 sehingga diperoleh Q1 untuk penjumlahan kolom c1, Q2 untuk penjumlahan kolom c2.

$$Q1 = 0,354509 + 0,173770 + 0,200405 + 0,386712 + 0,597699 + 0,818713 + 0,701820 + 0,719487 + 0,349173 + 0,851763 = 5,154050$$

$$Q2 = 0,170157 + 0,815438 + 0,190224 + 0,455563 + 0,526087 + 0,170134 + 0,231532 + 0,572203 + 0,101522 + 0,321522 = 3,554381$$

Langkah selanjutnya dilakukan proses normalisasi menggunakan Persamaan 2.5. Dimana setiap bilangan penyusun matriks awal dibagi dengan nilai Q sesuai dengan kolomnya masing-masing. Bilangan penyusun matriks disimbolkan dengan  $\mu_{ik}$  dimana i menunjukkan baris dan k menunjukkan kolom posisi bilangan tersebut pada matriks sehingga matriks partisi sudah mengalami normalisasi seperti yang ditunjukkan pada Tabel 4.8.

**Tabel 4.8** Derajat keanggotaan matriks partisi data A

i	$\mu$	
	C1	C2
1	0,068783	0,047872
2	0,033715	0,229418
3	0,038883	0,053518
4	0,075031	0,128169
5	0,115967	0,148011
6	0,158848	0,047866
7	0,136169	0,065140
8	0,139596	0,160985
9	0,067747	0,028562
10	0,165261	0,090458

d. Perhitungan Pusat *Cluster* (V)

Pada perhitungan pusat *cluster* (V) berdasarkan Persamaan 2.6. Setiap bilangan penyusun matriks partisi ( $\mu$ ) dipangkatkan dengan w, dimana variabel w yang sudah dideklarasikan terlebih dahulu dengan nilai sama dengan 2 dan dikalikan dengan bilangan penyusun matriks X yang merupakan matriks berisi data yang ada pada Tabel 4.1. Selanjutnya hasil perkalian tersebut dijumlah dan dibagi dengan jumlah hasil kuadrat dari bilangan penyusun matriks partisi. Hasil perhitungan ditunjukkan pada Tabel 4.9

**Tabel 4.9** Perhitungan pusat *cluster* data A

i	A	$\mu^2$		$\mu^2 \times X$	
		C1	C2	C1	C2
1	27533,00	0,004731	0,002292	130,263858	63,101138
2	20697,00	0,001137	0,052633	23,527257	1.089,365919
3	21588,00	0,001512	0,002864	32,639395	61,833796
4	23147,00	0,005630	0,016427	130,309523	380,246494
5	22240,00	0,013448	0,021907	299,090682	487,216379
6	27086,00	0,025233	0,002291	683,480254	62,060595
7	26118,00	0,018542	0,004243	484,278874	110,824708
8	23844,00	0,019487	0,025916	464,665668	617,965240

9	19441,00	0,004590	0,000816	89,231936	15,860815
10	20565,00	0,027311	0,008183	561,667812	168,280021
<b>Jumlah</b>		0,121620	0,137572	2.899,155259	3.056,755105

Berdasarkan jumlah *cluster* dari data A yaitu 2, maka terdapat 2 pusat *cluster* (V) yaitu V1, V2. Perhitungan nilai V atau pusat *cluster* untuk data A pada iterasi ke1 adalah 23837,758650 dan 22219,295122

e. Perhitungan Fungsi Objektif (P)

Pada perhitungan P menggunakan Persamaan 2.4, dimana setiap baris pada data matriks X dikurangi pusat *cluster* (V) selanjutnya dikuadratkan. Hasilnya kemudian dikalikan dengan  $\mu$  yang sudah dipangkatkan dengan bobot (w) seperti yang telah ditunjukkan pada Tabel 4.9. Selanjutnya hasil tersebut dijumlah per kolom seperti yang ditunjukkan pada Tabel 4.10.

**Tabel 4.10** Fungsi objektif data A

i	X	(X-V) <sup>2</sup>		(X-V) <sup>2</sup> x $\mu^2$	
		C1	C2	C1	C2
1	27533,00	13660647,73	28243855,81	64629,267	64728,445
2	20697,00	9860721,956	2315616,911	11208,834	121876,76
3	21588,00	5058939,552	397839,408	7648,5325	1139,489
4	23147,00	476967,932	860877,561	2685,1477	14141,955
5	22240,00	2552800,749	429,106	34330,871	9,400
6	27086,00	10557114,45	23693869,30	266386,01	54286,536
7	26118,00	5200002,277	15200757,45	96417,820	64500,054
8	23844,00	48,599	2642038,543	0,947	68471,648
9	19441,00	19324100,78	7714256,953	88691,549	6293,355
10	20565,00	10707611,22	2735005,229	292437,219	22379,544
<b>Jumlah</b>				864436,202	417827,197
<b>Fungsi Objektif (P)</b>				1282263,40036	

Perhitungan nilai fungsi objektif dilakukan seperti Persamaan 2.7, Nilai fungsi objektif diperoleh dengan cara sebagai berikut.

f. Normalisasi Matriks Partisi

Pada langkah ini normalisasi matriks partisi dilakukan berdasarkan Persamaan 2.8. Data matriks X untuk setiap barisnya dikurangi pusat *cluster* lalu dikuadratkan seperti yang ditunjukkan pada Tabel 4.10. Selanjutnya hasilnya

dipangkatkan dengan  $-1/1$  karena  $w-1 = 1$ . Hasil pangkat  $-1$  terhadap  $(X-V)^2$  ditunjukkan pada Tabel 4.11.

**Tabel 4.11** Normalisasi matriks partisi

i	$((X-V)^2)^{-1}$		Jumlah
	C1	C2	
1	0,000000073	0,000000035	0,000000109
2	0,000000101	0,000000432	0,000000533
3	0,000000198	0,000002514	0,000002711
4	0,000002097	0,000001162	0,000003258
5	0,000000392	0,002330426	0,002330817
6	0,000000095	0,000000042	0,000000137
7	0,000000192	0,000000066	0,000000258
8	0,020576252	0,000000378	0,020576630
9	0,000000052	0,000000130	0,000000181
10	0,000000093	0,000000366	0,000000459

Berdasarkan tabel diatas, bilangan penyusun matriks partisi yang telah diperbaiki diperoleh dari hasil pembagian data terhadap jumlah per barisnya. Pada perhitungan nilai matriks yang sudah diperbaiki dilakukan pada semua elemen matriks, sehingga hasil normalisasi matriks seperti yang ditunjukkan pada Tabel 4.12.

**Tabel 4.12** Hasil normalisasi matriks partisi

i	$\mu$	
	C1	C2
1	0,674005	0,325995
2	0,190173	0,809827
3	0,072907	0,927093
4	0,643481	0,356519
5	0,000168	0,999832
6	0,691772	0,308228
7	0,745107	0,254893
8	0,999982	0,000018
9	0,285308	0,714692

10	0,203458	0,796542
----	----------	----------

Setelah mendapatkan hasil normalisasi matriks partisi, kita lakukan pengecekan selama iterasi kurang dari iterasi maksimum (*maxIter*) atau selisih antara fungsi objektif iterasi saat ini dengan iterasi sebelumnya bernilai lebih dari nilai *error* terkecil yaitu 0,01 maka proses berlanjut dengan kembali melakukan perhitungan pusat *cluster* hingga normalisasi matriks partisi. Hasil perhitungan pusat *cluster* dari data A, B, C, dan D setelah dilakukan proses tiga kali iterasi dan diurutkan mulai dari yang terkecil sampai yang terbesar ditunjukkan pada Tabel 4.13.

**Tabel 4.13** Hasil pusat *cluster*

<i>Cluster</i>	A	B	C	D
1	26021,268248	78472,615679	33829,598997	20098,023017
2	21197,557572	79725,559510	33752,433566	20241,450858
3	-	-	34050,121528	-

#### 4.2.3.2 Perhitungan Batas Subinterval

Perhitungan batas subinterval dilakukan dengan mencari nilai tengah (median) dari setiap pusat *cluster*. Berikut merupakan contoh perhitungan batas subinterval pada data A:

- Interval 1  
 Batas bawah = 16580,96249 (Batas bawah *Universe of discourse* pada data A)  
 Batas atas = 23609,4129 (Median 21197,557572 dan 26021,268248)
- Interval 2  
 Batas bawah = 23609,4129 (Median 21197,557572 dan 26021,268248)  
 Batas atas = 30394,6675 (Batas atas *Universe of discourse* pada data A)

Seperti perhitungan diatas, juga dilakukan pada data produksi cabai (B), konsumsi cabai (C), dan luas panen (D). Dalam hal ini jumlah subinterval setiap atribut berbeda, sesuai dengan jumlah *cluster* yang sebelumnya ditunjukkan pada Tabel 4.6. Berikut merupakan batas subinterval masing–masing atribut yang ditunjukkan pada Tabel 4.14.

**Tabel 4.14** Batas subinterval masing–masing atribut data

<i>Interval</i>	A	B	C	D
u1	16580,9625	75804,7071	33226,8364	19946,6453
	s.d.	s.d.	s.d.	s.d.

	23609,4129	79099,0876	33791,0163	20169,7369
u2	23609,4129	79099,0876	33791,0163	20169,7369
	s.d.	s.d.	s.d.	s.d.
	30394,6675	82348,4529	33901,2775	20405,0447
u3			33901,2775	
	-	-	s.d.	-
			34366,0226	

#### 4.2.4 Pembentukan *Fuzzy Set*

Pembentukan *fuzzy set* berdasarkan pada derajat keanggotaan dari subinterval ( $\mu$ ). *fuzzy set* dari data A ditentukan dengan derajat keanggotaan sebagai berikut.

$$A1 = 1/\mu_1 + 0,5/\mu_2$$

$$A2 = 0,5/\mu_1 + 1/\mu_2$$

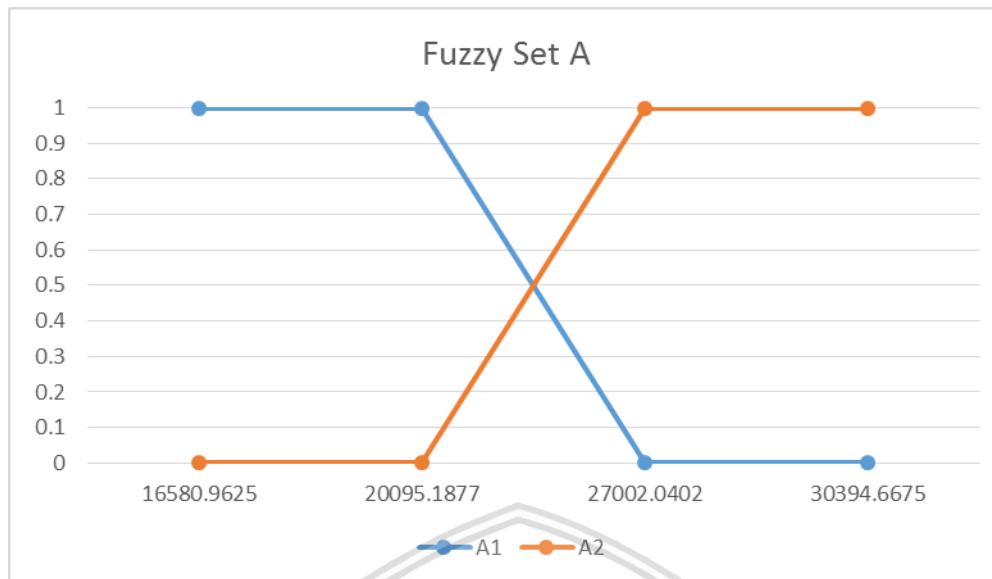
Pada derajat keanggotaan data A diatas, simbol “+” menunjukkan operator himpunan gabungan. Berdasarkan subinterval pada Tabel 4.14 nilai batas *fuzzy set* dari data atribut A dapat ditentukan dengan cara sebagai berikut.

$$A1 = 16580,9625 \text{ s.d. } 27002,04021 \text{ (Batas bawah interval 1 s.d. titik tengah interval 2)}$$

$$A2 = 20095,187698 \text{ s.d. } 30394,66751 \text{ (Titik tengah interval 1 s.d. batas atas interval 2)}$$

Setelah diketahui batas masing–masing *fuzzy set*, grafik fungsi keanggotaan *fuzzy set* data A yang direpresentasikan dalam kurva bahu yang ditunjukkan pada Gambar 4.22. Dengan cara yang sama juga ditentukan pula *fuzzy set* dari data atribut B, C, dan D.





**Gambar 4.22** Grafik *fuzzy set* data A

#### 4.2.5 Fuzzifikasi

Pada tahap fuzzifikasi, dilakukan dengan mencari derajat keanggotaan yang memiliki nilai paling besar (maksimum) dari masing-masing data terhadap *fuzzy set* yang sudah terbentuk. Derajat keanggotaan dicari berdasarkan fungsi keanggotaan kurva bentuk bahu. Berikut ini adalah perhitungan derajat keanggotaan dari data atribut A:

Data aktual harga cabai (A) pada periode Februari 2012= 20697,00

$$A1 = \frac{27002,04021 - 20697,00}{27002,04021 - 20095,187698} = 0,912783$$

$$A2 = \frac{20697,00 - 20095,187698}{27002,04021 - 20095,187698} = 0,087217$$

Derajat keanggotaan paling besar berada pada A1 sehingga data atribut A pada Februari 2012 difuzzifikasikan ke *fuzzy set* A1. Perhitungan yang sama dilakukan pada seluruh data atribut A sehingga didapat derajat keanggotaan dan hasil fuzzifikasi dari data A ditunjukkan pada Tabel 4.15.

**Tabel 4.15** Hasil derajat keanggotaan dan *fuzzy set* data A

i	A	1	2	Fuzzifikasi
1	27533,00	0	1	A2
2	20697,00	0,912783	0,087217	A1
3	21588,00	0,783785	0,216215	A1
4	23147,00	0,558128	0,441872	A1

5	22240,00	0,689465	0,310535	A1
6	27086,00	0,000000	1,000000	A2
7	26118,00	0,127979	0,872021	A2
8	23844,00	0,457127	0,542873	A2
9	19441,00	1,000000	0,000000	A1
10	20565,00	0,931904974	0,068095026	A1

Perhitungan yang sama juga dilakukan pada semua attribut yaitu attribut B, C, dan D. Sehingga dapat diperoleh hasil fuzzifikasi sesuai derajat keanggotaan dengan nilai paling besar. Berikut adalah hasil fuzzifikasi untuk semua attribut yang dapat dilihat pada Tabel 4.16.

**Tabel 4.16** Hasil fuzzifikasi semua data attribut

No.	Periode	A	B	C	D
1	Januari 2012	A2	B1	C1	D1
2	Februari 2012	A1	B1	C2	D1
3	Maret 2012	A1	B1	C2	D1
4	April 2012	A1	B1	C3	D1
5	Mei 2012	A1	B1	C3	D1
6	Juni 2012	A2	B2	C3	D1
7	Juli 2012	A2	B2	C3	D2
8	Agustus 2012	A2	B2	C3	D2
9	September 2012	A1	B2	C2	D2
10	Januari 2012	A1	B2	C2	D2

#### 4.2.6 Pembentukan *Fuzzy Logical Relationship* (FLR)

Pada tahap pembentukan FLR dengan berdasarkan nilai *order* yang sudah ditentukan. Pada manualisasi ini berdasarkan dari Tabel 4.16 di misalkan nilai *order* kita deklarasikan dengan nilai 3 sehingga dapat menghasilkan 7 FLR. Hasil pembentukan FLR Data *training* dapat ditunjukkan pada Tabel 4.17. Hasil pembentukan FLR data *testing* dapat ditunjukkan pada Tabel 4.18.

**Tabel 4.17** Hasil pembentukan FLR data *training*

No.	<i>Fuzzy Logical Relationship</i> (FLR)
1	$(A_{(1,2)}, B_{(1,1)}, C_{(1,1)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,2)}, D_{(2,1)}), (A_{(3,1)}, B_{(3,1)}, C_{(3,2)}, D_{(3,1)}) \rightarrow A_1$
2	$(A_{(1,1)}, B_{(1,1)}, C_{(1,2)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,2)}, D_{(2,1)}), (A_{(3,1)}, B_{(3,1)}, C_{(3,3)}, D_{(3,1)}) \rightarrow A_1$
3	$(A_{(1,1)}, B_{(1,1)}, C_{(1,2)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,3)}, D_{(2,1)}), (A_{(3,1)}, B_{(3,1)}, C_{(3,3)}, D_{(3,1)}) \rightarrow A_2$
4	$(A_{(1,1)}, B_{(1,1)}, C_{(1,3)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,3)}, D_{(2,1)}), (A_{(3,2)}, B_{(3,2)}, C_{(3,3)}, D_{(3,1)}) \rightarrow A_2$
5	$(A_{(1,1)}, B_{(1,1)}, C_{(1,3)}, D_{(1,1)}), (A_{(2,2)}, B_{(2,2)}, C_{(2,3)}, D_{(2,1)}), (A_{(3,2)}, B_{(3,2)}, C_{(3,3)}, D_{(3,2)}) \rightarrow A_2$

**Tabel 4.18** Hasil pembentukan FLR data *testing*

No.	<i>Fuzzy Logical Relationship</i> (FLR)
1	$(A_{(1,2)}, B_{(1,2)}, C_{(1,3)}, D_{(1,1)}), (A_{(2,2)}, B_{(2,2)}, C_{(2,3)}, D_{(2,2)}), (A_{(3,2)}, B_{(3,2)}, C_{(3,3)}, D_{(3,2)}) \rightarrow A_1$
2	$(A_{(1,2)}, B_{(1,2)}, C_{(1,3)}, D_{(1,2)}), (A_{(2,2)}, B_{(2,2)}, C_{(2,3)}, D_{(2,2)}), (A_{(3,1)}, B_{(3,2)}, C_{(3,2)}, D_{(3,2)}) \rightarrow A_1$

#### 4.2.7 Defuzzifikasi

Pada tahap ini, misalkan jumlah data yang digunakan untuk data uji atau data *testing* adalah dua yaitu, yang pertama data periode ke-9 dan yang kedua data periode ke-10. Sehingga yang digunakan data latih atau data *training* adalah data pada periode ke-1 sampai data periode ke-8. Pada data *testing* yang pertama melibatkan FLR ke-1 sampai FLR ke-5 untuk disesuaikan dengan FLR ke-6 yang berdasarkan pada Tabel 4.17. Pada data *testing* yang kedua melibatkan FLR ke-1 sampai FLR ke-6 untuk disesuaikan dengan FLR ke-7 yang berdasarkan pada Tabel 4.17.

Selanjutnya setiap ruas kiri FLR data *testing* dibandingkan dengan FLR data *training* yaitu, pada bilangan pangkat bawah *fuzzy set* dibandingkan sesuai dengan posisinya dan dihitung selisihnya. Hasil dari perhitungan selisih tersebut dijumlahkan, dan apabila jumlah selisihnya kurang dari *threshold* maka FLR

tersebut dikatakan cocok. *Threshold* merupakan dari hasil mengalikan jumlah *order* dengan jumlah faktor kemudian dijumlahkan dengan dengan sebuah konstanta. Contoh perhitungan selisih nilai ruas kiri FLR pada data *testing* pertama adalah sebagai berikut.

Data *testing* =

$(A_{(1,2)}, B_{(1,2)}, C_{(1,3)}, D_{(1,1)}), (A_{(2,2)}, B_{(2,2)}, C_{(2,3)}, D_{(2,2)}), (A_{(3,2)}, B_{(3,2)}, C_{(3,3)}, D_{(3,2)}) \rightarrow A_1$

Data *training* =

1.  $(A_{(1,2)}, B_{(1,1)}, C_{(1,1)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,2)}, D_{(2,1)}), (A_{(3,1)}, B_{(3,1)}, C_{(3,2)}, D_{(3,1)}) \rightarrow A_1$

2.  $(A_{(1,1)}, B_{(1,1)}, C_{(1,2)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,2)}, D_{(2,1)}), (A_{(3,1)}, B_{(3,1)}, C_{(3,3)}, D_{(3,1)}) \rightarrow A_1$

3.  $(A_{(1,1)}, B_{(1,1)}, C_{(1,2)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,3)}, D_{(2,1)}), (A_{(3,1)}, B_{(3,1)}, C_{(3,3)}, D_{(3,1)}) \rightarrow A_2$

4.  $(A_{(1,1)}, B_{(1,1)}, C_{(1,3)}, D_{(1,1)}), (A_{(2,1)}, B_{(2,1)}, C_{(2,3)}, D_{(2,1)}), (A_{(3,2)}, B_{(3,2)}, C_{(3,3)}, D_{(3,1)}) \rightarrow A_2$

5.  $(A_{(1,1)}, B_{(1,1)}, C_{(1,3)}, D_{(1,1)}), (A_{(2,2)}, B_{(2,2)}, C_{(2,3)}, D_{(2,1)}), (A_{(3,2)}, B_{(3,2)}, C_{(3,3)}, D_{(3,2)}) \rightarrow A_2$

- Selisih data ke-1 =  $|2-2| + |2-1| + |3-1| + |1-1| + |2-1| + |2-1| + |3-2| + |2-1| + |2-1| + |2-1| + |3-2| + |2-1| = 11$
- Selisih data ke-2 =  $|2-1| + |2-1| + |3-2| + |1-1| + |2-1| + |2-1| + |3-2| + |2-1| + |2-1| + |2-1| + |3-3| + |2-1| = 10$
- Selisih data ke-3 =  $|2-1| + |2-1| + |3-2| + |1-1| + |2-1| + |2-1| + |3-3| + |2-1| + |2-2| + |2-2| + |3-3| + |2-1| = 4$
- Selisih data ke-4 =  $|2-1| + |2-1| + |3-3| + |1-1| + |2-1| + |2-1| + |3-3| + |2-1| + |2-2| + |2-2| + |3-3| + |2-1| = 6$
- Selisih data ke-5 =  $|2-1| + |2-1| + |3-3| + |1-1| + |2-2| + |2-2| + |3-3| + |2-1| + |2-2| + |2-2| + |3-3| + |2-2| = 3$

Pada perhitungan selisih nilai ruas kiri FLR pada data *testing* juga dilakukan pada semua data *testing* yang sudah dideklarasikan sebelumnya. Dalam hal ini, agar dapat menemukan FLR yang cocok dengan data *testing*. Tahap selanjutnya adalah proses defuzzifikasi menggunakan *threshold* yang dimisalkan 18, sehingga FLR yang memiliki kecocokan dengan data *testing* pertama adalah semua FLR dari ke-1 sampai ke-5 karena memiliki hasil selisih kurang dari 18. Semua FLR yang cocok ditampung, dikelompokkan berdasarkan FLR pada ruas kanan yang sama kemudian dihitung frekuensinya. Berikut adalah frekuensi dan pusat *cluster* dari ruas kanan FLR yang cocok dengan data *testing* pertama dan kedua ditunjukkan pada Tabel 4.19.

**Tabel 4.19** Frekuensi FLR yang cocok dengan data *testing*

	<b>Data Testing 1</b>		<b>Data Testing 2</b>	
<b>Ruas Kanan</b>	A1	A2	A1	A2
<b>Pusat Cluster</b>	21197,5575	26021,2682	21197,5575	26021,2682
<b>Frekuensi</b>	3	2	3	3

Berdasarkan pada Tabel 4.19, dilakukan defuzzifikasi sesuai dengan Persamaan 2.18 sebagai berikut.

$$\text{Data Testing 1} = \frac{(21197,5575 \times 3) + (26021,2682 \times 2)}{5} = 24091,7840$$

$$\text{Data Testing 1} = \frac{(21197,5575 \times 3) + (26021,2682 \times 3)}{6} = 23609,4129$$

Berdasarkan perhitungan diatas, diperoleh hasil peramalan harga cabai pada pada sedangkan pada periode Bulan Oktober 2012 adalah 23609,4129. Perbandingan data aktual harga cabai dan hasil peramalan dengan metode *High Order Fuzzy Times Series Multifactors* dapat dilihat pada Tabel 4.20.

**Tabel 4.20** Perbandingan data aktual dan hasil peramalan

<b>No.</b>	<b>Data Aktual</b>	<b>Hasil Peramalan</b>
1	19441,00	24091,7840
2	20565,00	23609,4129

Dari hasil peramalan yang ditunjukkan pada Tabel 4.20 dapat dilakukan perhitungan kesalahan peramalan menggunakan metode *Mean Square Error* (MSE) seperti Persamaan 2.19.

$$MSE = \frac{((19441,00 - 24091,7840)^2 + (20565,00 - 23609,4129)^2)}{2}$$

$$MSE = 26254651,46$$

Nilai MSE untuk hasil perhitungan kesalahan peramalan menunjukkan nilai 26254651,46. Nilai tersebut mempunyai basis jutaan, Hal tersebut dikarenakan data yang digunakan dalam peramalan merupakan data harga (Rp). Untuk mengubahnya menjadi data aktual maka dilakukan penyederhanaan dengan membaginya, sehingga nilai MSE yang dihasilkan sebesar 26254,65146.

### 4.3 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan dari tampilan antarmuka atau *interface* yang nantinya akan diterapkan pada tahapan implementasi. Pada perancangan antarmuka ini, terdapat lima halaman utama dan juga terdapat dalam menu berupa *button* yaitu halaman input, halaman tampil data, halaman proses, halaman hasil, dan halaman perhitungan *error*.

#### 4.3.1 Halaman Input

Pada halaman input terdapat masukan data dengan nama *button Browse Data*, pada masukan data ini pengguna diharuskan memasukkan *file* data dalam bentuk ekstensi *.xls* serta direktori lokasi file yang diinputkan akan ditampilkan pada *textfield*. Pada bagian ini juga terdapat label yang menampilkan jumlah banyaknya data yang terdapat pada file yang telah di inputkan. Selanjutnya juga terdata inputan *textfield* data *training*, data *testing*, *order* dan nilai konstanta. Pada halaman input ini juga terdapat *button fuzzification* yang berfungsi untuk melakukan proses *clustering*, fuzzifikasi, dan FLR. Selanjutnya pada *button forecasting* berfungsi untuk melakukan peramalan atau defuzzifikasi dari hasil proses fuzzifikasi serta menghitung nilai *error* dari hasil peramalan. Berikut merupakan rancangan tampilan antarmuka dari halaman input yang ditunjukkan pada Gambar 4.23.

The diagram illustrates the layout of the input page interface, enclosed in a large rectangular frame. The layout consists of 13 numbered boxes arranged as follows:

- Box 1:** A large horizontal rectangle at the top of the frame.
- Box 2:** A small rectangle located below Box 1 on the left side.
- Box 3:** A small rectangle located below Box 1, to the right of Box 2.
- Box 4:** A small rectangle located below Box 1, to the right of Box 3.
- Box 5:** A small rectangle located below Box 1, to the right of Box 4.
- Box 6:** A small rectangle located below Box 1 on the right side.
- Box 7:** A small rectangle located below Box 2.
- Box 8:** A horizontal rectangle located below Box 7.
- Box 9:** A horizontal rectangle located below Box 8.
- Box 10:** A horizontal rectangle located below Box 9.
- Box 11:** A horizontal rectangle located below Box 10.
- Box 12:** A small rectangle located below Box 7 and to the left of Box 13.
- Box 13:** A small rectangle located below Box 11 and to the right of Box 12.

**Gambar 4.23** Rancangan antarmuka halaman input

Penjelasan pada rancangan antarmuka halaman input ditunjukkan pada Tabel 4.21.

**Tabel 4.21** Keterangan rancangan antarmuka halaman input

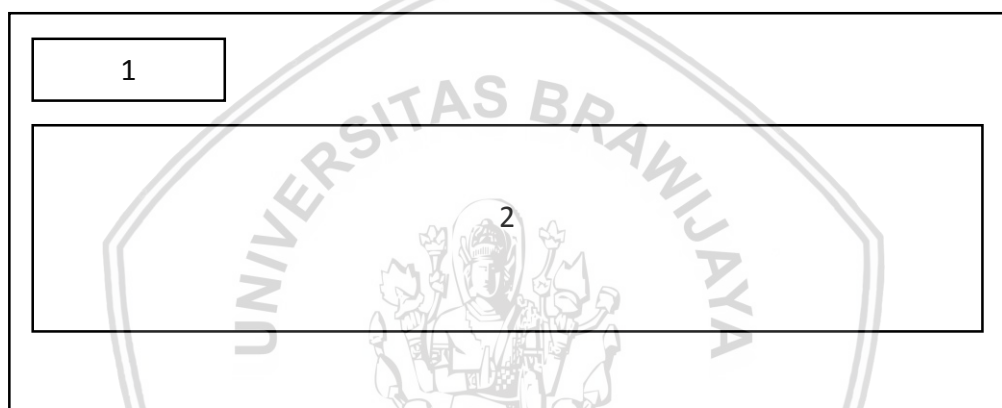
No.	Nama objek	Tipe	Keterangan
1	Nama program	<i>Textarea</i>	Merupakan nama program yang dibuat
2	Input data	<i>Tab and Button</i>	Merupakan tab untuk melihat tampilan input data
3	Tampil data	<i>Tab and Button</i>	Merupakan tab untuk melihat tampilan tampil data
4	Proses	<i>Tab and Button</i>	Merupakan tab untuk melihat tampilan proses
5	Hasil	<i>Tab and Button</i>	Merupakan tab untuk melihat tampilan hasil peramalan
6	Perhitungan <i>error</i>	<i>Tab and Button</i>	Merupakan tab untuk melihat tampilan perhitungan error
7	<i>Browse data</i>	<i>Button and Textfield</i>	Merupakan tombol untuk mencari data (.xls) pada komputer
8	Data traning	<i>Textfield</i>	Merupakan input jumlah data <i>training</i>
9	Data <i>testing</i>	<i>Textfield</i>	Merupakan input jumlah data <i>testing</i>
10	<i>Order</i>	<i>Textfield</i>	Merupakan input nilai <i>order</i>
11	Konstanta	<i>Textfield</i>	Merupakan input nilai konstanta
12	<i>Fuzzification</i>	<i>Button</i>	Merupakan tombol melakukan proses fuzzifikasi



13	<i>Forecasting</i>	<i>Button</i>	Merupakan tombol melakukan proses peramalan
----	--------------------	---------------	---

### 4.3.2 Halaman Tampil Data

Pada halaman tampil data terdapat tabel yang menampilkan isi data dari file excel yang sudah dimasukkan terlebih dahulu pada halaman input. Tabel tampil data akan periode, harga cabai, jumlah produksi, jumlah konsumsi dan luas lahan panen. Berikut merupakan tampilan rancangan halaman tampil data yang ditunjukkan pada Gambar 4.24.



**Gambar 4.24** Rancangan antarmuka halaman tampil data

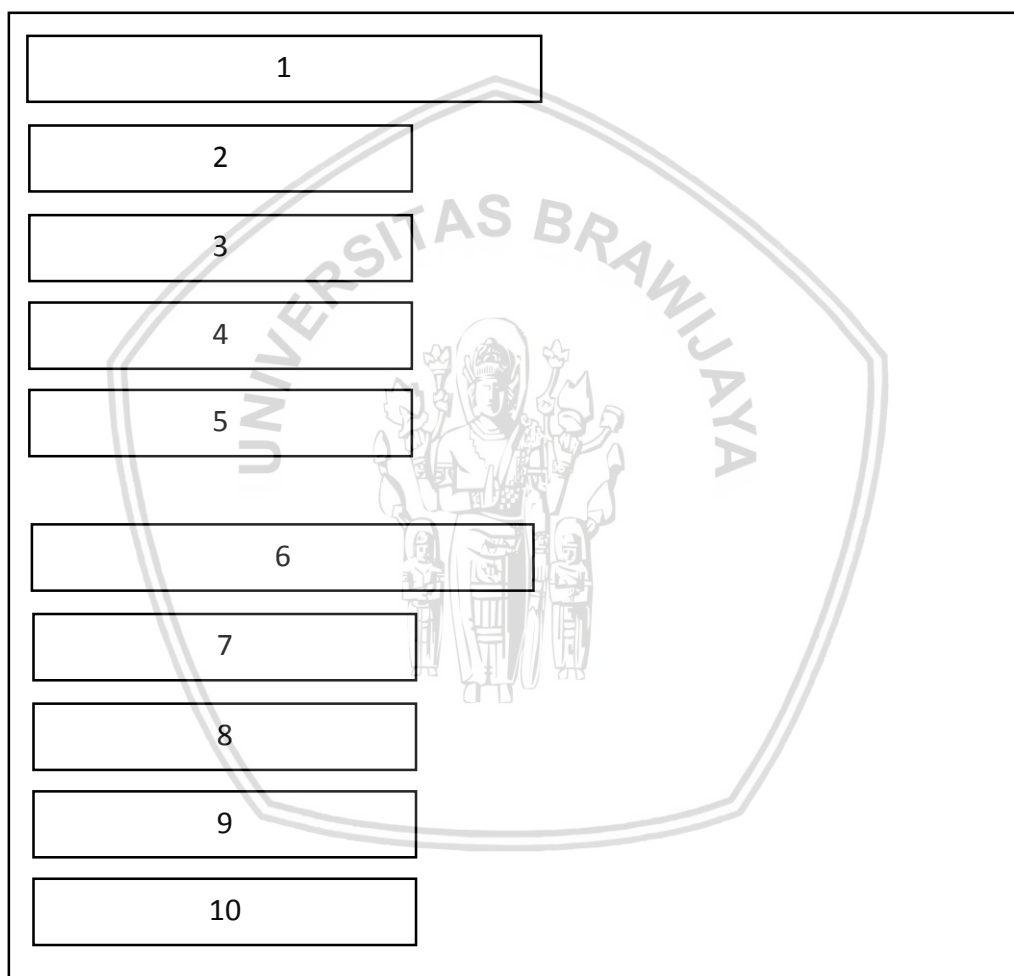
Penjelasan pada rancangan antarmuka halaman tampil data ditunjukkan pada Tabel 4.22.

**Tabel 4.22** Keterangan rancangan antarmuka halaman tampil data

No.	Nama objek	Tipe	Keterangan
1	Tampil data	<i>Textarea</i>	Merupakan nama tampilan antarmuka
2	Tabel data	Tabel	Merupakan tabel untuk menampilkan data yang digunakan sebagai data peramalan

### 4.3.3 Halaman Proses

Pada halaman proses akan menampilkan hasil *clustering*, fuzzifikasi, dan *fuzzy logical relationship*. Pada tampilan halaman *clustering* akan menampilkan *Universe of discourse* harga cabai, jumlah produksi, jumlah konsumsi dan luas lahan panen. Selanjutnya tampilan dibawahnya akan menampilkan jumlah *cluster* dari setiap faktor yaitu jumlah *cluster* harga cabai, jumlah produksi, jumlah konsumsi dan luas lahan panen. Berikut merupakan tampilan perancangan halaman proses *clustering* yang ditunjukkan Gambar 4.25.



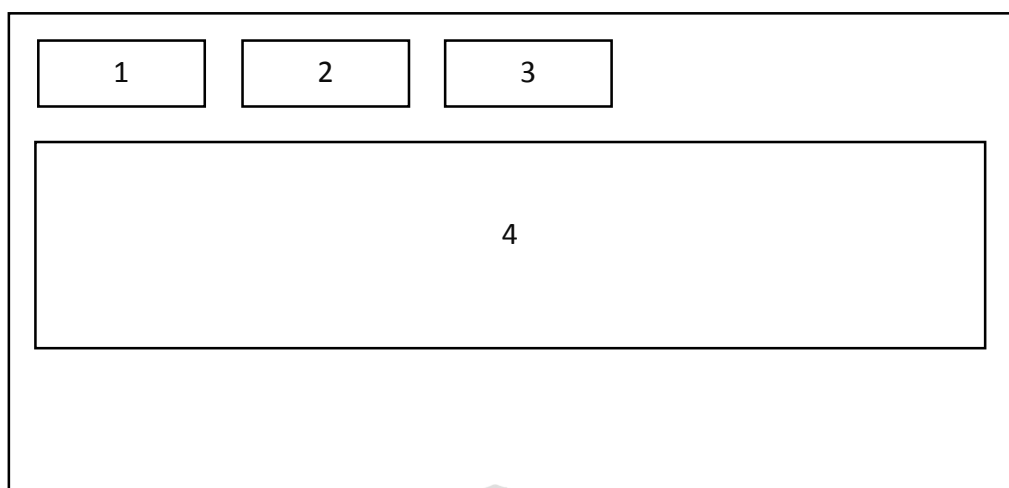
**Gambar 4.25** Rancangan antarmuka halaman proses *clustering*

Penjelasan pada rancangan antarmuka halaman proses *clustering* ditunjukkan pada Tabel 4.23.

**Tabel 4.23** Keterangan rancangan antarmuka halaman proses *clustering*

No.	Nama objek	Tipe	Keterangan
1	<i>Universe of Discourse</i>	<i>Textarea</i>	Merupakan proses <i>universe of discourse</i>
2	<i>Universe of Discourse</i> harga cabai	<i>Textfield</i>	Merupakan hasil proses <i>universe of discourse</i> harga cabai
3	<i>Universe of Discourse</i> jumlah produksi	<i>Textfield</i>	Merupakan hasil proses <i>universe of discourse</i> jumlah produksi
4	<i>Universe of Discourse</i> jumlah konsumsi	<i>Textfield</i>	Merupakan hasil proses <i>universe of discourse</i> jumlah produksi
5	<i>Universe of Discourse</i> luas lahan	<i>Textfield</i>	Merupakan hasil proses <i>universe of discourse</i> luas lahan
6	Jumlah <i>cluster</i> setiap faktor	<i>Textarea</i>	Merupakan proses menentukan jumlah <i>cluster</i> setiap faktor
7	Jumlah <i>cluster</i> harga cabai	<i>Textfield</i>	Merupakan jumlah <i>cluster</i> harga cabai
8	Jumlah <i>cluster</i> jumlah produksi	<i>Textfield</i>	Merupakan jumlah <i>cluster</i> jumlah produksi
9	Jumlah <i>cluster</i> jumlah konsumsi	<i>Textfield</i>	Merupakan jumlah <i>cluster</i> jumlah konsumsi
10	Jumlah <i>cluster</i> luas lahan	<i>Textfield</i>	Merupakan jumlah <i>cluster</i> luas lahan

Selanjutnya pada tampilan proses fuzzifikasi akan menampilkan tabel hasil fuzzifikasi dari harga cabai, jumlah produksi, jumlah konsumsi dan luas lahan panen. Berikut adalah tampilan rancangan proses fuzzifikasi yang ditunjukkan pada Gambar 4.26.



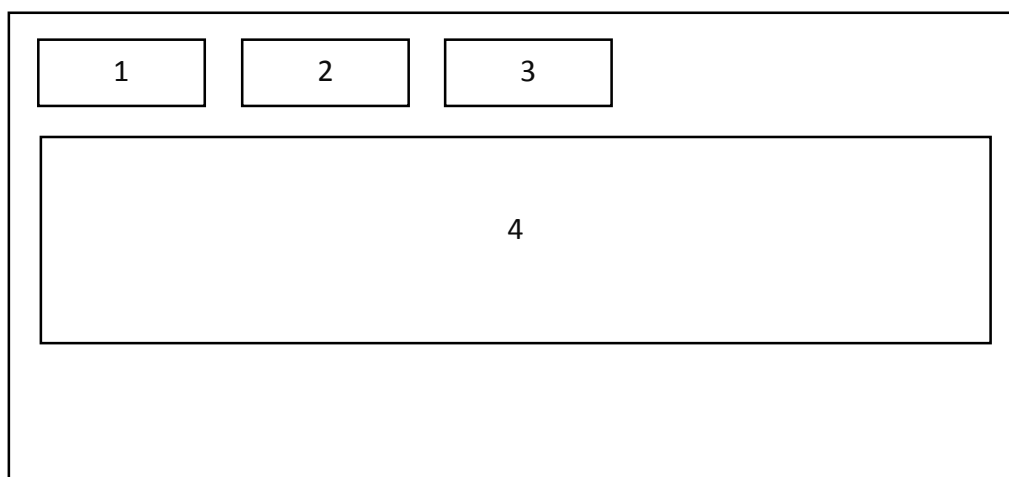
**Gambar 4.26** Rancangan antarmuka proses fuzzifikasi

Penjelasan pada rancangan antarmuka proses fuzzifikasi ditunjukkan pada Tabel 4.24.

**Tabel 4.24** Keterangan rancangan antarmuka proses fuzzifikasi

No.	Nama objek	Tipe	Keterangan
1	<i>Clustering</i>	Tab	Merupakan tab untuk menampilkan proses clustering
2	<i>Fuzzification</i>	Tab	Merupakan tab untuk menampilkan proses fuzzifikasi
3	<i>Fuzzy Logical Relationship</i>	Tab	Merupakan tab untuk menampilkan proses <i>fuzzy logical relationship</i>
4	Tabel Hasil Fuzzifikasi	Tabel	Merupakan tabel untuk menampilkan nilai hasil fuzzifikasi empat faktor peramalan

Pada tampilan proses pembetulan *fuzzy logical relationship* (FLR) akan menampilkan *textarea* berisi hasil proses FLR dari harga cabai, jumlah produksi, jumlah konsumsi dan luas lahan panen. Berikut adalah tampilan rancangan proses *fuzzy logical relationship* yang ditunjukkan pada Gambar 4.27.



**Gambar 4.27** Rancangan antarmuka proses *fuzzy logical relationship* (FLR)

Penjelasan pada rancangan antarmuka *proses fuzzy logical relationship* ditunjukkan pada Tabel 4.25.

**Tabel 4.25** Keterangan rancangan antarmuka proses *fuzzy logical relationship*

No.	Nama objek	Tipe	Keterangan
1	<i>Clustering</i>	Tab	Merupakan tab untuk menampilkan proses <i>clustering</i>
2	<i>Fuzzification</i>	Tab	Merupakan tab untuk menampilkan proses fuzzifikasi
3	<i>Fuzzy logical relationship</i>	Tab	Merupakan tab untuk menampilkan proses <i>fuzzy logical relationship</i>
4	Tabel hasil <i>fuzzy logical relationship</i>	<i>Textfield</i>	Merupakan tabel untuk menampilkan nilai hasil pembentukan <i>fuzzy logical relationship</i>

#### 4.3.4 Halaman Hasil

Pada halaman hasil peramalan akan menampilkan tabel yang berisi hasil peramalan harga cabai dan data harga cabai yang sebenarnya. Pada halaman ini terdapat *textlink* yang akan menampilkan grafik perbandingan data hasil peramalan dan data yang sebenarnya. Berikut adalah tampilan rancangan halaman hasil yang ditunjukkan pada Gambar 4.28.

1		
2	3	4

**Gambar 4.28** Rancangan antarmuka halaman hasil peramalan

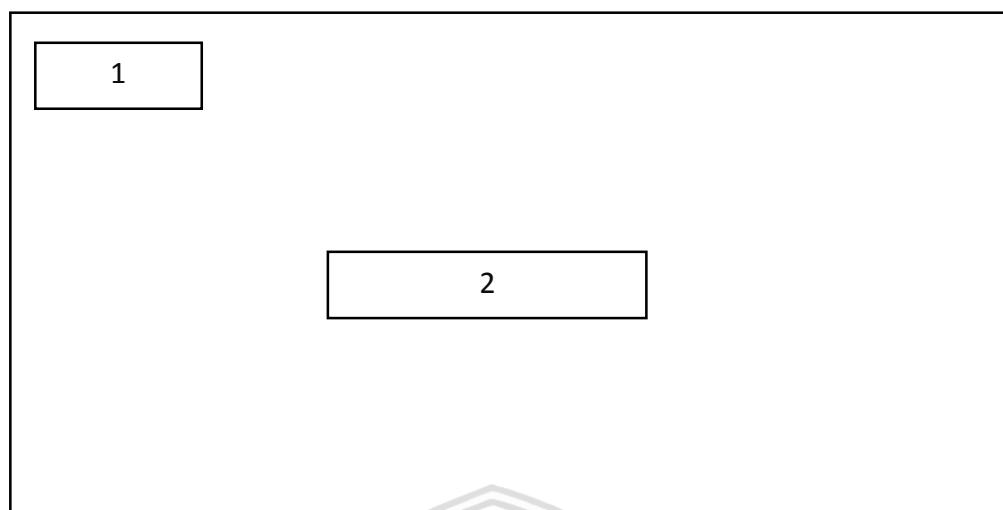
Penjelasan pada rancangan antarmuka halaman hasil peramalan ditunjukkan pada Tabel 4.26.

**Tabel 4.26** Keterangan rancangan antarmuka halaman hasil peramalan

No.	Nama objek	Tipe	Keterangan
1	Hasil peramalan	<i>Textarea</i>	Merupakan nama tampilan antarmuka
2	<i>Title 1</i>	<i>Textfield</i>	Merupakan tabel untuk menampilkan periode data yang diramalkan
3	<i>Title 2</i>	<i>Textfield</i>	Merupakan tabel untuk menampilkan nilai data aktual
4	<i>Title 3</i>	<i>Textfield</i>	Merupakan tabel untuk menampilkan nilai hasil peramalan

#### 4.3.5 Halaman Perhitungan *Error*

Pada halaman perhitungan *error* akan menampilkan nilai dari hasil perhitungan *Mean Square Error* (MSE) pada *textfield*. Tampilan rancangan halaman perhitungan *error* yang ditunjukkan pada Gambar 4.29.



**Gambar 4.29** Rancangan antarmuka halaman perhitungan *error*

Penjelasan pada rancangan antarmuka halaman perhitungan *error* ditunjukkan pada Tabel 4.27.

**Tabel 4.27** Keterangan rancangan antarmuka halaman perhitungan *error*

No.	Nama objek	Tipe	Keterangan
1	Perhitungan <i>error</i>	<i>Textarea</i>	Merupakan nama tampilan antarmuka
2	Tombol MSE	<i>Button and Textfield</i>	Merupakan tombol untuk menampilkan nilai MSE



## 4.4 Perancangan Pengujian

Pada penelitian ini akan dilakukan empat pengujian yaitu pengujian terhadap nilai *order*, nilai *threshold*, pengujian terhadap jumlah data *training* untuk mengetahui pengaruhnya terhadap nilai *Mean Square Error* (MSE), dan pengujian validasi sistem untuk mengetahui kestabilan hasil implementasi.

### 4.4.1 Perancangan Pengujian Pengaruh *Order* Terhadap MSE

*Order* merupakan jumlah urutan data yang digunakan dalam pembentukan *Fuzzy Logical Relationship* (FLR). Pada pengujian ini dilakukan percobaan dengan *order* yang berbeda-beda. Pengujian ini bertujuan untuk mengetahui pengaruh *order* terhadap hasil peramalan. Dalam setiap pengujiannya akan dilakukan sebanyak sepuluh kali selanjutnya nilai MSE yang diperoleh dihitung rata-ratanya. Pada setiap pengujian yang telah dilakukan dihitung nilai MSE untuk mengetahui tingkat kesalahan metode ini ketika menggunakan nilai *order* yang berbeda. Berikut adalah perancangan pengujian pengaruh *order* terhadap nilai MSE ditunjukkan pada Tabel 4.28.

**Tabel 4.28** Rancangan pengujian pengaruh *order* terhadap MSE

No.	<i>Order</i>	<i>Threshold</i>	MSE
1	....	....	....
...	....	....	....
....	....	....	....
...	....	....	....
....	....	....	....

### 4.4.2 Perancangan Pengujian Pengaruh Data *Training* Terhadap MSE

Data *training* berperan penting dalam proses defuzzifikasi, pada penelitian ini dilakukan pengujian pengaruh jumlah data *training* terhadap tingkat kesalahan dari hasil peramalan. Pada pengujian ini dilakukan percobaan dengan jumlah data *training* yang berbeda-beda. Dalam setiap pengujiannya dilakukan sebanyak sepuluh kali selanjutnya nilai MSE yang diperoleh dihitung rata-ratanya. Dari setiap percobaan dilakukan perhitungan nilai MSE untuk mengetahui tingkat kesalahan dari hasil peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* dengan jumlah data *training* yang berbeda-beda. Sehingga hasil pengujian ini akan diketahui tingkat pengaruh dari jumlah data *training* yang digunakan. Berikut merupakan rancangan pengujian pengaruh jumlah data *training* yang ditunjukkan pada Tabel 4.29.

**Tabel 4.29** Rancangan pengujian pengaruh data *training* terhadap MSE

No.	Jumlah Data <i>Training</i>	MSE
1	....	....
...	....	....
....	....	....
...	....	....
....	....	....

#### 4.4.4 Perancangan Validasi Sistem

Pada pengujian validasi sistem akan melakukan pengujian dengan menggunakan *order*, *threshold* terbaik seperti yang telah dilakukan pada pengujian–pengujian sebelumnya yang menghasilkan hasil *error* MSE terendah. Dalam setiap pengujiannya akan dilakukan sebanyak sepuluh kali selanjutnya nilai MSE yang diperoleh dihitung rata–ratanya. Rancangan pengujian validasi sistem yang ditunjukkan pada Tabel 4.30.

**Tabel 4.30** Rancangan pengujian validasi sistem

Percobaan ke-i	MSE
1	....
...	....
....	....
...	....
....	....

## BAB 5 IMPLEMENTASI

Pada bab ini berisi pembahasan mengenai hasil implementasi metode *High Order Fuzzy Times Series Multifactors* sesuai dengan bab sebelumnya. Pada bab implementasi terdiri dari spesifikasi sistem, implementasi algoritme, dan implementasi antarmuka.

### 5.1 Spesifikasi Sistem

Pada spesifikasi sistem ini dibagi mejadi dua yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

#### 5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras merupakan perangkat keras yang digunakan pada implementasi penelitian ini. Perangkat keras yang digunakan adalah laptop dengan spesifikasi sesuai dengan pada Tabel 5.1.

**Tabel 5.1** Spesifikasi perangkat keras

Komponen	Spesifikasi
CPU	CPU AMD A4-3305M
GPU	AMD Radeon™ HD 6480G
Memori (RAM)	DDR3 2 GB
Harddisk	500 GB

#### 5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak merupakan perangkat lunak yang digunakan pada implementasi penelitian ini. Perangkat lunak yang digunakan pada implementasi penelitian sesuai dengan yang ditunjukkan pada Tabel 5.2.

**Tabel 5.2** Spesifikasi perangkat lunak

Sistem Operasi	Windows 10 Home Premium 64-bit
Bahasa Pemrograman	Java
Tools Pemrograman	Netbeans IDE 8.0
Browser	Mozilla Firefox versi 59.0.1
Dokumentasi Penelitian	Microsoft Word 2013
Manualisasi Penelitian	Microsoft Excel 2013

## 5.2 Implementasi Algoritme

Implementasi algoritme membahas mengenai hasil implementasi metode *High Order Fuzzy Times Series Multifactors* yang mengacu pada bab sebelumnya yaitu perancangan. Dalam penelitian ini implementasi algoritme menggunakan bahasa pemrograman java berbasis desktop. Implementasi algoritme terdiri dari beberapa method sesuai dengan langkah–langkah pada algoritme metode *High Order Fuzzy Times Series Multifactors*.

### 5.2.1 Implementasi Proses *Universe of Discourse*

Pada implementasi *Universe of Discourse* ditunjukkan pada Kode Program 5.1. Baris 2–4 : Terdapat pemanggilan method *mean*, simpangan\_baku, dan *Dmin\_dan\_Dmax*. Pemanggilan method tersebut dibutuhkan untuk melakukan proses perhitungan selanjutnya yaitu untuk menghitung batas bawah dan batas atas dari *Universe of Discourse*.

Baris 7–10 : Yaitu fungsi untuk menghitung hasil *Universe of Discourse*. Berikut merupakan kode program dari proses *Universe of Discourse* yang ditunjukkan pada Kode Program 5.1.

Baris	Kode
1	<code>private void Universe_of_Discourse() {</code>
2	<code>    mean();</code>
3	<code>    standar_deviasi();</code>
4	<code>    Min_dan_Max();</code>
5	<code>    hsl_uMin = new double[jmlh_Faktor];</code>
6	<code>    hsl_uMax = new double[jmlh_Faktor];</code>
7	<code>    for (int i = 0; i &lt; jmlh_Faktor; i++) {</code>
8	<code>        hsl_uMin[i] = hsl_min[i] - hsl_stdev[i];</code>
9	<code>        hsl_uMax[i] = hsl_max[i] + hsl_stdev[i];</code>
10	<code>    }</code>
11	<code>}</code>

Kode Program 5.1 Implementasi proses *universe of discourse*

#### 5.2.1.1 Implementasi Proses *Mean*

Implementasi proses *mean* merupakan proses perhitungan nilai rata–rata dari setiap faktor yang nantinya hasilnya diperlukan untuk proses perhitungan simpangan baku. Berikut merupakan Kode Program dari proses *mean* yang ditunjukkan pada Kode Program 5.2. Perhitungan utama proses *mean* terdapat pada baris 3–8.

Baris	Kode
1	<code>private void mean() {</code>
2	<code>    hsl_mean = new double[jmlh_Faktor];</code>
3	<code>    for (int i = 0; i &lt; jmlh_Faktor; i++) {</code>
4	<code>        double tmp_jmlh = 0;</code>
5	<code>        for (int j = 0; j &lt; jmlh_DataTraining; j++) {</code>
6	<code>            tmp_jmlh = tmp_jmlh + data[i][j];</code>
7	<code>        }</code>
8	<code>        hsl_mean[i] = (tmp_jmlh / jmlh_DataTraining);</code>
9	<code>    }</code>
10	<code>}</code>

**Kode Program 5.2** Implementasi proses *mean*

### 5.2.1.2 Implementasi Proses Simpangan Baku

Implementasi proses perhitungan simpangan baku merupakan perhitungan akar kuadrat dari *varians*. Simpangan baku juga merupakan ukuran keragaman data statistik untuk menentukan bagaimana sebaran data dalam sampel. Simpangan baku dapat dilakukan perhitungan setelah mendapat nilai hasil dari perhitungan *mean*. Berikut merupakan *source code* dari implementasi proses simpangan baku yang ditunjukkan pada Kode Program 5.3. Perhitungan utama proses simpangan baku terdapat pada baris 3–9.

Baris	Kode
1	<code>private void standar_deviasi() {</code>
2	<code>    hsl_stdev = new double[jmlh_Faktor];</code>
3	<code>    for (int i = 0; i &lt; jmlh_Faktor; i++) {</code>
4	<code>        double result = 0;</code>
5	<code>        for (int j = 0; j &lt; jmlh_DataTraining; j++) {</code>
6	<code>            result=result+(Math.pow((data[i][j]-hsl_mean[i]),</code>
7	<code>                2));</code>
8	<code>        }</code>
9	<code>        hsl_stdev[i]=Math.sqrt(result/(jmlh_DataTraining-1));</code>
10	<code>    }</code>
11	<code>}</code>

**Kode Program 5.3** Implementasi proses simpangan baku

### 5.2.1.3 Implementasi Proses *Dmin* dan *Dmax*

Implementasi proses *Dmin* dan *Dmax* merupakan perhitungan untuk mencari nilai minimum dan nilai maksimum dari data sampel pada setiap faktor. Berikut merupakan Kode Program dari implementasi dari proses *Dmin* dan *Dmax* yang ditunjukkan pada Kode Program 5.4.

Baris 8–10 : Merupakan bagian dari proses untuk untuk menentukan nilai data terbesar.

Baris 11–13 : Merupakan proses ntuk menentukan nilai data terkecil.

Baris	Kode
1	<code>private void Min_dan_Max() {</code>
2	<code>    hsl_max = new double[jmlh_Faktor];</code>
3	<code>    hsl_min = new double[jmlh_Faktor];</code>
4	<code>    for (int i = 0; i &lt; jmlh_Faktor; i++) {</code>
5	<code>        double besar = data[i][0];</code>
6	<code>        double kecil = data[i][0];</code>
7	<code>        for (int j = 0; j &lt; jmlh_DataTraining; j++) {</code>
8	<code>            if (data[i][j] &gt; besar) {</code>
9	<code>                besar = data[i][j];</code>
10	<code>            }</code>
11	<code>            if (data[i][j] &lt; kecil) {</code>
12	<code>                kecil = data[i][j];</code>
13	<code>            }</code>
14	<code>        }</code>
15	<code>        hsl_max[i] = besar;</code>
16	<code>        hsl_min[i] = kecil;</code>
17	<code>    }</code>
18	<code>}</code>

**Kode Program 5.4** Implementasi proses *Dmin* dan *Dmax*

### 5.2.2 Implementasi Proses Jumlah *Cluster*

Implementasi proses jumlah *cluster* merupakan proses perhitungan jumlah *cluster* dari setiap faktor. Implementasi ini dapat dilakukan apabila sudah mendapatkan hasil perhitungan dari method *Dmin\_dan\_Dmax()*. Berikut merupakan kode program dari implementasi proses jumlah *cluster* yang ditunjukkan pada Kode Program 5.5.

Baris 3–6 : Digunakan untuk menghitung selisih data sampel setiap baris dari semua faktor pada data sampel ke *t* dengan *t*+1.

Baris 8–9 : Untuk menghitung jumlah *cluster* dari setiap faktor.

Baris	Kode
1	<code>private void jumlah_Cluster() {</code>
2	<code>    jmlh_Cluster = new int[jmlh_Faktor];</code>
3	<code>    for (int i = 0; i &lt; jmlh_Faktor; i++) {</code>
4	<code>        double selisih = 0;</code>
5	<code>        for (int j = 0; j &lt; (jmlh_DataTraining - 1); j++) {</code>
6	<code>            selisih = selisih + (Math.abs((data[i][j]-data[i][(j</code>
7	<code>                + 1)]));</code>
8	<code>        }</code>
9	<code>        jmlh_Cluster[i] = (int) Math.round((Math.abs(hsl_min[i]-</code>
10	<code>            hsl_max[i]))/(selisih/(jmlh_DataTraining-1)));</code>
11	<code>    }</code>
12	<code>}</code>

**Kode Program 5.5** Implementasi proses jumlah *cluster*

### 5.2.3 Implementasi Proses Pusat *Cluster*

Implementasi proses perhitungan pusat *cluster* ini menggunakan metode *fuzzy C-means* untuk mendapatkan pusat *cluster* yang nantinya digunakan sebagai subinterval. Implementasi ini memerlukan hasil dari proses method jumlah\_Cluster() agar dapat melakukan perhitungan subinterval. Berikut



merupakan kode program dari implementasi proses subinterval yang ditunjukkan pada Kode Program 5.6. Kode Program 5.6 hanya menampilkan source code pembentukan subinterval pada faktor Harga (A) karena dari Jumlah Produksi (B), Jumlah Konsumsi (C), dan Luas Lahan Panen (D) memiliki kode program yang sama namun terjadi perubahan nama variabel.

Baris 2–3 : Merupakan deklarasi variabel bobot, iterasi maksimum, selisih fungsi objektif awal, dan *error* minimum.

Baris 8–20 : Merupakan proses pembentukan matriks partisi dengan nilai random 0 sampai 1.

Baris 22–26 : Merupakan proses normalisasi matriks partisi yang telah terbentuk.

Baris 28–63 : Merupakan proses untuk menentukan pusat *cluster*.

Baris 65–86 : Merupakan proses perhitungan fungsi objektif dari iterasi.

Baris 88–90 : Merupakan proses perhitungan selisih fungsi objektif.

Baris 92–108 : Adalah proses normalisasi matriks partisi.

Baris 110–115 : Adalah proses hasil normalisasi matriks partisi.

Baris 116–133 : Adalah proses dengan pengkondisian apakah selisih fungsi objektif lebih kecil dari *error* minimum serta apabila memenuhi kondisi maka dilanjutkan ke proses pengurutan pusat *cluster* secara *ascending* dari hasil iterasi terakhir dan proses pengkondisian iterasi berhenti.

Baris	Kode
1	private void pusat_cluster() {
2	int bobot = 2; int iter_maksimum = 100;
3	double selisih_f_objektif = 0; double error_min = 0.01;
4	for (int i = 0; i < jmlh_Faktor; i++) {
5	//faktor A
6	if (i == 0) {
7	//Pembentukan matriks partisi
8	tmp_Matriks1 = new
9	double[jmlh_Cluster[i]][jmlh_DataTraining];
10	q1 = new double[jmlh_Cluster[i]];
11	double jmlh_q1;
12	for (int j = 0; j < jmlh_Cluster[i]; j++) {
13	jmlh_q1 = 0;
14	for (int k = 0; k < jmlh_DataTraining; k++) {
15	double random = ThreadLocalRandom.current().
16	nextDouble(0.1, 1);
17	tmp_Matriks1[j][k] = random;
18	jmlh_q1 = jmlh_q1 + random;
19	}
20	q1[j] = jmlh_q1;
21	}
22	//normalisasi matriks
23	for (int j = 0; j < jmlh_Cluster[i]; j++) {
24	for (int k = 0; k < jmlh_DataTraining; k++) {
25	tmp_Matriks1[j][k] = tmp_Matriks1[j][k] / q1[j];
26	}
27	}
28	//Perhitungan pusat cluster
29	for (int iter = 0; iter < iter_maksimum; iter++) {
30	selisih_f_objektif = 0;
31	jumA1 = new double[iter_maksimum][jmlh_Cluster[i]];



```

32     jumA2 = new double[iter_maksimum][jmlh_Cluster[i]];
33     miu_Matriks1 = new double[jmlh_Cluster[i]]
34         [jmlh_DataTraining];
35     XV_Matriks1 = new double[jmlh_Cluster[i]]
36         [jmlh_DataTraining];
37     pusat_cluster1 = new double[jmlh_Cluster[i]];
38     double hitung_jum1;
39     double hitung_jum2;
40     for (int j = 0; j < jmlh_Cluster[i]; j++) {
41         hitung_jum1 = 0;
42         for (int k = 0; k < jmlh_DataTraining; k++) {
43             tmp_Matriks1[j][k] = Math.pow(tmp_Matriks1
44                 [j][k], bobot);
45             miu_Matriks1[j][k] = tmp_Matriks1[j][k];
46             hitung_jum1 = hitung_jum1 + tmp_Matriks1
47                 [j][k];
48         }
49         jumA1[iter][j] = hitung_jum1;
50     }
51     for (int j = 0; j < jmlh_Cluster[i]; j++) {
52         hitung_jum2 = 0;
53         for (int k = 0; k < jmlh_DataTraining; k++) {
54             tmp_Matriks1[j][k] = tmp_Matriks1[j][k]
55                 * data[i][k];
56             hitung_jum2 = hitung_jum2 +
57                 tmp_Matriks1[j][k];
58         }
59         jumA2[iter][j] = hitung_jum2;
60     }
61     for (int j = 0; j < jmlh_Cluster[i]; j++) {
62         pusat_cluster1[j] = jumA2[iter][j] /
63             jumA1[iter][j];
64     }
65     //perhitungan fungsi objektif
66     jumA3 = new double[iter_maksimum][jmlh_Cluster[i]];
67     for (int j = 0; j < jmlh_Cluster[i]; j++) {
68         double hitung_jum3 = 0;
69         for (int k = 0; k < jmlh_DataTraining; k++) {
70             tmp_Matriks1[j][k] = Math.pow((data[i][k]
71                 - pusat_cluster1[j]),
72                 bobot);
73             XV_Matriks1[j][k] = tmp_Matriks1[j][k];
74             tmp_Matriks1[j][k] = XV_Matriks1[j][k]
75                 * miu_Matriks1[j][k];
76             hitung_jum3 = hitung_jum3 +
77                 tmp_Matriks1[j][k];
78         }
79         jumA3[iter][j] = hitung_jum3;
80     }
81     double jmlh_f_objektif = 0;
82     for (int j = 0; j < jmlh_Cluster[i]; j++) {
83         jmlh_f_objektif = jmlh_f_objektif
84             + jumA3[iter][j];
85     }
86     if (iter < (iter_maksimum - 1)) {
87         f_objektif1[iter + 1] = jmlh_f_objektif;
88         //perhitungan selisih fungsi objektif
89         selisih_f_objektif = f_objektif1[iter + 1]
90             - f_objektif1[iter];
91     }
92     //normalisasi matriks partisi
93     int hitung = bobot - 1; int pangkat = -1 / hitung;
94     for (int j = 0; j < jmlh_Cluster[i]; j++) {
95         for (int k = 0; k < jmlh_DataTraining; k++) {
96             tmp_Matriks1[j][k] = Math.pow(XV_Matriks1

```

```

97         [j][k], pangkat);
98     }
99 }
100 jumA4 = new
101 double[iter_maksimum][jmlh_DataTraining];
102 for (int j = 0; j < jmlh_DataTraining; j++) {
103     double hitung_jum4 = 0;
104     for (int k = 0; k < jmlh_Cluster[i]; k++) {
105         hitung_jum4 = hitung_jum4 +
106             tmp_Matriks1[k][j];
107     }
108     jumA4[iter][j] = hitung_jum4;
109 }
110 //hasil normalisasi matriks
111 for (int j = 0; j < jmlh_Cluster[i]; j++) {
112     for (int k = 0; k < jmlh_DataTraining; k++) {
113         tmp_Matriks1[j][k] = tmp_Matriks1[j][k] /
114             jumA4[iter][k];
115     }
116 }
117 if (selisih_f_objektif < error_min) {
118     //pengurutan ascending pusat cluster
119     for (int j = 0; j < jmlh_Cluster[i]; j++) {
120         for (int k = 0; k < jmlh_Cluster[i]; k++) {
121             if (k < (jmlh_Cluster[i] - 1)) {
122                 if (pusat_cluster1[k] >
123                     pusat_cluster1[k + 1]) {
124                     double tmp =
125                         pusat_cluster1[k];
126                     pusat_cluster1[k] =
127                         pusat_cluster1[k + 1];
128                     pusat_cluster1[k + 1] = tmp;
129                 }
130             }
131         }
132     }
133     iter = iter_maksimum;
134 }
135 }
136 }

```

Kode Program 5.6 Implementasi proses pusat *cluster*

#### 5.2.4 Implementasi Proses Pembentukan Subinterval dan *Fuzzy Set*

Implementasi proses pembentukan subinterval dan *fuzzy set* memiliki dua proses yaitu proses menentukan batas subinterval minimum dan maksimum serta proses menentukan batas *fuzzy set* minimum dan maksimum. Proses pembentukan *fuzzy* dapat menjalankan prosesnya setelah method subinterval() telah diakses. Proses pembentukan *fuzzy set* ditunjukkan pada Kode Program 5.7. Pada Kode Program 5.7 hanya menampilkan kode program pembentukan *fuzzy set* pada faktor Harga (A) karena dari faktor Jumlah Produksi (B), Jumlah Konsumsi (C), dan Luas Lahan Panen (D) memiliki kode program yang sama namun hanya terjadi perubahan nama variabel.

Baris 6–29 : Merupakan proses pembentukan batas minimum dan batas maksimum subinterval.

Baris 31–54 : Merupakan pembentukan batas minimum dan batas maksimum

*fuzzy set.*

Baris	Kode
1	private void subinterval_dan_fuzzy_set() {
2	for (int i = 0; i < jmlh_Faktor; i++) {
3	//faktor A
4	if (i == 0) {
5	//batas subinterval
6	batas_subintervalMinA = new double[jmlh_Cluster[i]];
7	batas_subintervalMaxA = new double[jmlh_Cluster[i]];
8	for (int j = 0; j < jmlh_Cluster[i]; j++) {
9	if (j == 0) {
10	batas_subintervalMinA[j]=hsl_uMin[i];
11	batas_subintervalMaxA[j]=(pusat_cluster1[j]
12	+pusat_cluster1[j+
13	1))/2;
14	} else if (j == (jmlh_Cluster[i] - 1)) {
15	batas_subintervalMinA[j]=(pusat_cluster1[j-
16	1]+pusat_cluster1
17	[j])/2;
18	batas_subintervalMaxA[j]=
19	hsl_uMax[i];
20	} else if ((0 < j) && (j < (jmlh_Cluster[i]-1))) {
21	batas_subintervalMinA[j]=(pusat_cluster1[j-1]+
22	pusat_cluster1[j])/2;
23	batas_subintervalMaxA[j]=(pusat_cluster1[j]+
24	pusat_cluster1[j+1])/2;
25	}
26	}
27	//batas fuzzy set
28	batas_fuzzysetMinA = new double[jmlh_Cluster[i]];
29	batas_fuzzysetMaxA = new double[jmlh_Cluster[i]];
30	for (int j = 0; j < jmlh_Cluster[i]; j++) {
31	if (j == 0) {
32	batas_fuzzysetMinA[j]=
33	batas_subintervalMinA[j];
34	batas_fuzzysetMaxA[j]=
35	(batas_subintervalMinA[j+1]
36	+batas_subintervalMaxA[j + 1])/2;
37	} else if (j == (jmlh_Cluster[i] - 1)) {
38	batas_fuzzysetMinA[j]=
39	(batas_subintervalMinA[j-1]
40	+batas_subintervalMaxA[j - 1])/2;
41	batas_fuzzysetMaxA[j] =
42	batas_subintervalMaxA[j];
43	} else if ((0 < j) && (j < (jmlh_Cluster[i]-1))) {
44	batas_fuzzysetMinA[j]=
45	(batas_subintervalMinA[j-1]+
46	batas_subintervalMaxA[j-1])/2;
47	batas_fuzzysetMaxA[j]=
48	batas_subintervalMinA[j+1]+
49	batas_subintervalMaxA[j + 1])/2;
50	}
51	}
52	}
53	}

**Kode Program 5.7** Implementasi proses pembentukan subinterval dan *fuzzy set*

### 5.2.5 Implementasi Proses Fuzzifikasi

Implementasi proses fuzzifikasi memiliki dua proses utama yaitu menentukan derajat keanggotaan dan proses fuzzifikasi. Dalam proses ini method `fuzzy_set()` harus dieksekusi terlebih dahulu karena implementasi proses ini memerlukan hasil dari method tersebut. Kode program fuzzifikasi ditunjukkan pada Kode Program 5.8. Kode Program 5.8 hanya menampilkan kode pembentukan *fuzzy set* pada faktor Harga (A) karena dari Jumlah Produksi (B), Jumlah Konsumsi (C), dan Luas Lahan Panen (D) memiliki kode program yang sama namun hanya terjadi perubahan nama variabel.

Baris 3–81 : Merupakan proses perhitungan derajat keanggotaan.

Baris 83–94 : Adalah proses penentuan fuzzifikasi.

Baris	Kode
1	<code>private void fuzzifikasi() {</code>
2	<code>    if (i == 0) {</code>
3	<code>        hitung_fuzzifikasiA = new double[jmlh_Cluster[i]]</code>
4	<code>        [jmlh_DataTraining];</code>
5	<code>        for (int j = 0; j &lt; jmlh_Cluster[i]; j++) {</code>
6	<code>            for (int k = 0; k &lt; jmlh_DataTraining; k++) {</code>
7	<code>                if(j == 0) {</code>
8	<code>                    if((data[i][k]&gt;batas_fuzzysetMinA[j])&amp;&amp;</code>
9	<code>                        (data[i][k]&lt;batas_fuzzysetMinA[j+1])) {</code>
10	<code>                        hitung_fuzzifikasiA[j][k] = 1;</code>
11	<code>                    }</code>
12	<code>                    if ((data[i][k] &gt; batas_fuzzysetMinA[j+1])</code>
13	<code>                        &amp;&amp; (data[i][k] &lt; batas_fuzzysetMaxA[j])) {</code>
14	<code>                        { hitung_fuzzifikasiA[j][k]=</code>
15	<code>                            (batas_fuzzysetMaxA[j] - data[i][k]) /</code>
16	<code>                            (batas_fuzzysetMaxA[j] -</code>
17	<code>                                batas_fuzzysetMinA[j + 1]);</code>
18	<code>                        }</code>
19	<code>                    if (!(data[i][k] &gt;batas_fuzzysetMinA[j])</code>
20	<code>                        &amp;&amp;!(data[i][k]&lt;batas_fuzzysetMinA[j+1])</code>
21	<code>                        &amp;&amp;!(data[i][k]&gt;batas_fuzzysetMinA[j+1])</code>
22	<code>                        &amp;&amp;!(data[i][k]&lt;batas_fuzzysetMaxA[j])){</code>
23	<code>                        hitung_fuzzifikasiA[j][k] = 0;</code>
24	<code>                    }</code>
25	<code>                }</code>
26	<code>            } else if (j == (jmlh_Cluster[i] - 1)) {</code>
27	<code>                if ((data[i][k] &gt; batas_fuzzysetMinA</code>
28	<code>                    [(jmlh_Cluster[i] - 1)]) &amp;&amp;</code>
29	<code>                    (data[i][k] &lt; batas_fuzzysetMaxA</code>
30	<code>                    [(jmlh_Cluster[i] - 2)])) {</code>
31	<code>                    hitung_fuzzifikasiA[j][k]=</code>
32	<code>                        (data[i][k]-batas_fuzzysetMinA</code>
33	<code>                        [(jmlh_Cluster[i] - 1)])/</code>
34	<code>                        (batas_fuzzysetMaxA[(jmlh_Cluster[i]-</code>
35	<code>                            2)]-batas_fuzzysetMinA</code>
36	<code>                            [(jmlh_Cluster[i] - 1)]));</code>
37	<code>                }</code>
38	<code>                if ((data[i][k] &gt; batas_fuzzysetMaxA</code>
39	<code>                    [(jmlh_Cluster[i]- 2)]) &amp;&amp; (data[i][k]</code>
40	<code>                    &lt; batas_fuzzysetMaxA[(jmlh_Cluster[i]</code>
41	<code>                        - 1)])) {</code>
42	<code>                    hitung_fuzzifikasiA[j][k] = 1;</code>
43	<code>                }</code>
44	<code>            } if (!(data[i][k] &gt; batas_fuzzysetMinA</code>

```

45         [(jmlh_Cluster[i] - 1))] &&
46         !(data[i][k] < batas_fuzzysetMaxA
47         [(jmlh_Cluster[i] - 2))] &&
48         !(data[i][k] > batas_fuzzysetMaxA
49         [(jmlh_Cluster[i] - 2))] &&
50         !(data[i][k] < batas_fuzzysetMaxA
51         [(jmlh_Cluster[i] - 1)])) {
52             hitung_fuzzifikasiA[j][k] = 0;
53         }
54     }
55     else if ((0 < j)&&(j < (jmlh_Cluster[i]-1))) {
56         if ((data[i][k] > batas_fuzzysetMinA[j])
57             && (data[i][k] < batas_fuzzysetMinA[j
58             +1])) {
59             hitung_fuzzifikasiA[j][k]=
60             (data[i][k]-batas_fuzzysetMinA[j])
61             /(batas_fuzzysetMinA[j + 1]-
62             batas_fuzzysetMinA[j]);
63         }
64         if ((data[i][k] > batas_fuzzysetMinA
65             [j+1])&& (data[i][k] <
66             batas_fuzzysetMaxA[j])) {
67             hitung_fuzzifikasiA[j][k]=
68             (batas_fuzzysetMaxA[j]-data[i][k])/
69             (batas_fuzzysetMaxA[j]-
70             batas_fuzzysetMinA[j + 1]);
71         }
72         if (!(data[i][k] > batas_fuzzysetMinA[j])
73             && !(data[i][k] < batas_fuzzysetMinA
74             [j + 1]) && !(data[i][k] >
75             batas_fuzzysetMinA[j + 1]) &&
76             !(data[i][k]<batas_fuzzysetMaxA[j])){
77             hitung_fuzzifikasiA[j][k] = 0;
78         }
79     }
80 }
81 }
82 hasil_fuzzifikasiA = new int[jmlh_DataTraining];
83 for (int j = 0; j < jmlh_DataTraining; j++) {
84     int cluster = 0;
85     double besar=hitung_fuzzifikasiA[0][j];
86     for (int k = 0; k < jmlh_Cluster[i]; k++) {
87         if (besar < hitung_fuzzifikasiA[k][j]) {
88             besar = hitung_fuzzifikasiA[k][j];
89             cluster = k;
90         }
91     }
92     hasil_fuzzifikasiA[j] = cluster + 1;
93 }
94 }
95 }

```

Kode Program 5.8 Implementasi proses fuzzifikasi

### 5.2.6 Implementasi Pembentukan *Fuzzy Logical Relationship*

Implementasi pembentukan *Fuzzy Logical Relationship* (FLR) memiliki dua *output* yaitu FLR\_kiri dan FLR\_kanan. Dimana FLR\_kiri berisi *group* dari kelima faktor dan pada FLR\_kanan merupakan dari faktor yang akan diramalkan yaitu faktor Harga (A). Berdasarkan dari perancangan proses pada bab sebelumnya,

maka implementasi kode program pembentukan *Fuzzy Logical Relationship* (FLR) dapat ditunjukkan pada Kode Program 5.9.

Baris 3–14 : Merupakan deklarasi variabel.

Baris 15–55 : Adalah proses pembentukan *fuzzy logical relationship* data *training*.

Baris 56–58 : Merupakan deklarasi variabel.

Baris 59–86 : yaitu proses pembentukan *fuzzy logical relationship* data *testing*.

Baris	Kode
1	//Fuzzy Logical Relationship
2	private void fuzzy_logical_relationship() {
3	String input_order = fieldOrder.getText();
4	order = Integer.parseInt(input_order);
5	int max_inc = order;
6	String input_dataTraining = fieldDatatraining.getText();
7	int dataTraining = Integer.valueOf(input_dataTraining);
8	int tmp_jmlhDatatraining = 0;
9	int tmp_order = 0;
10	int jmlh_dataTraining = dataTraining - order;
11	String input_dataTesting = fieldDatatesting.getText();
12	int jmlh_dataTesting = Integer.valueOf(input_dataTesting);
13	FLR_kiri_training = new
14	int[jmlh_DataTraining][order][jmlh_Faktor];
15	FLR_kanan_training = new int[jmlh_DataTraining];
16	for (int i = 0; i < jmlh_dataTraining; i++) {
17	for (int j = i; j < max_inc; j++) {
18	for (int k = 0; k < jmlh_Faktor; k++) {
19	if (j == 0) {
20	if (k == 0) {
21	FLR_kiri_training[i][j][k]=
22	hasil_fuzzifikasiA[j];
23	} else if (k == 1) {
24	FLR_kiri_training[i][j][k] =
25	hasil_fuzzifikasiB[j];
26	} else if (k == 2) {
27	FLR_kiri_training[i][j][k] =
28	hasil_fuzzifikasiC[j];
29	} else if (k == 3) {
30	FLR_kiri_training[i][j][k] =
31	hasil_fuzzifikasiD[j];
32	}
33	} else if (j != 0) {
34	if (k == 0) {
35	FLR_kiri_training[i][j - i][k] =
36	hasil_fuzzifikasiA[j];
37	} else if (k == 1) {
38	FLR_kiri_training[i][j - i][k] =
39	hasil_fuzzifikasiB[j];
40	} else if (k == 2) {
41	FLR_kiri_training[i][j - i][k] =
42	hasil_fuzzifikasiC[j];
43	} else if (k == 3) {
44	FLR_kiri_training[i][j - i][k] =
45	hasil_fuzzifikasiD[j];
46	}
47	}
48	}
49	tmp_order = j;
50	}
51	FLR_kanan_training[i] = hasil_fuzzifikasiA[(i + order)];
52	if (max_inc < (jmlh_DataTraining - 1)) {



```

53         max_inc = max_inc + 1;
54     }
55     tmp_jmlhDatatraining = i;
56 }
57 FLR_kiri_testing = new
58 int[jmlh_DataTraining][order][jmlh_Faktor];
59 FLR_kanan_testing = new int[(jmlh_DataTraining)];
60 tmp_jmlhDatatraining = tmp_jmlhDatatraining + 1;
61 for (int i = 0; i < jmlh_dataTesting; i++) {
62     for (int j = tmp_jmlhDatatraining; j < max_inc; j++) {
63         for (int k = 0; k < jmlh_Faktor; k++) {
64             if (k == 0) {
65                 FLR_kiri_testing[i][j -
66                     tmp_jmlhDatatraining][k]
67                     = hasil_fuzzifikasiA[j];
68             } else if (k == 1) {
69                 FLR_kiri_testing[i][j -
70                     tmp_jmlhDatatraining][k] =
71                     hasil_fuzzifikasiB[j];
72             } else if (k == 2) {
73                 FLR_kiri_testing[i][j -
74                     tmp_jmlhDatatraining][k] =
75                     hasil_fuzzifikasiC[j];
76             } else if (k == 3) {
77                 FLR_kiri_testing[i][j -
78                     tmp_jmlhDatatraining][k] =
79                     hasil_fuzzifikasiD[j];
80             }
81         }
82     }
83     FLR_kanan_testing[i] =
84     hasil_fuzzifikasiA[(tmp_jmlhDatatraining + order)];
85     if (max_inc < (jmlh_DataTraining - 1)) {
86         max_inc = max_inc + 1;
87     }
88     tmp_jmlhDatatraining++;
89 }
90 }

```

**Kode Program 5.9** Implementasi proses pembentukan *fuzzy logical relationship*

### 5.2.7 Implementasi Proses Defuzzifikasi

Implementasi proses defuzzifikasi merupakan proses menentukan hasil peramalan dari data *testing* berdasarkan dari bilangan *fuzzy* yang terbentuk. Implementasi proses defuzzifikasi memiliki beberapa masukan penting yaitu data *training*, data *testing*, *order*, dan konstanta. Masukan nilai tersebut sangat berperan penting dalam implementasi proses defuzzifikasi. Berdasarkan dari perancangan proses pada bab sebelumnya maka implementasi kode program defuzzifikasi ditunjukkan pada Kode Program 5.10.

Baris 2–11 : Merupakan deklarasi variabel.

baris 13–37 : Merupakan proses perhitungan selisih antara data *training* dengan data *testing*.

baris 40–47 : Merupakan proses menghitung jumlah FLR yang cocok.

baris 49–65 : Merupakan proses mencocokkan FLR kiri yang cocok dengan FLR kanan.

baris 67–81 : Merupakan proses menghitung jumlah frekuensi FLR kanan yang



cocok.

baris 83–91 : Adalah proses perhitungan hasil peramalan.

Baris	Kode
1	private void defuzifikasi() {
2	String input_dataTraining = fieldDatatraining.getText();
3	String input_dataTesting = fieldDatatesting.getText();
4	String input_konstanta = fieldKonstanta.getText();
5	int dataTraining = Integer.valueOf(input_dataTraining);
6	int dataTesting = Integer.valueOf(input_dataTesting);
7	int konstanta = Integer.valueOf(input_konstanta);
8	int match[][];
9	match = new int[dataTesting][2000];
10	int threshold = (jmlh_Faktor * order) + konstanta;
11	int hitung_selisih = 0;
12	//perhitungan selisih
13	for (int i = 0; i < dataTesting; i++) {
14	for (int j = 0; j < (dataTraining + i); j++) {
15	hitung_selisih = 0;
16	for (int k = 0; k < order; k++) {
17	for (int l = 0; l < jmlh_Faktor; l++) {
18	if (i == 0) {
19	hitung_selisih = hitung_selisih +
20	(Math.abs(FLR_kiri_training[j][k][l] -
21	FLR_kiri_testing[i][k][l]));
22	} else if (i > 0) {
23	FLR_kiri_training[(dataTraining-
24	1)+i][k][l] = FLR_kiri_testing[i-
25	1][k][l];
26	FLR_kanan_training[(dataTraining-1)+i]
27	= FLR_kanan_testing[i-1];
28	hitung_selisih = hitung_selisih +
29	(Math.abs(FLR_kiri_training[j][k][l] -
30	FLR_kiri_testing[i][k][l]));
31	}
32	}
33	}
34	if (hitung_selisih < threshold) {
35	match[i][j] = hitung_selisih;
36	}
37	}
38	}
39	//menghitung jumlah total FLR yang cocok untuk setiap data
40	testing
41	int countGroup[] = new int[dataTesting];
42	for (int i = 0; i < dataTesting; i++) {
43	for (int j = 0; j < (dataTraining + i); j++) {
44	if (match[i][j] > 0) {
45	countGroup[i]++;
46	}
47	}
48	}
49	//mencocokkan FLR_kiri yang cocok dengan FLR_kanan
50	group = new int[dataTesting][2000];
51	for (int i = 0; i < dataTesting; i++) {
52	int tmp_count = 0;
53	for (int j = 0; j < (dataTraining + i); j++) {
54	if (match[i][j] > 0) {
55	if (tmp_count < countGroup[i]) {
56	group[i][tmp_count] = j;
57	tmp_count++;
58	}
59	}

```

60     }
61 }
62 for (int i = 0; i < dataTesting; i++) {
63     for (int j = 0; j < countGroup[i]; j++) {
64         group[i][j] = FLR_kanan[group[i][j]];
65     }
66 }
67 //mengelompokkan yang sama
68 Map<Integer, Integer> map = new HashMap<Integer,Integer>();
69 hasil_peramalan = new double[dataTesting];
70 for (int i = 0; i < dataTesting; i++) {
71     double peramalan = 0;
72     map.clear();
73     for (int j = 0; j < countGroup[i]; j++) {
74         int key = group[i][j];
75         if (map.containsKey(key)) {
76             int occurrence = map.get(key);
77             occurrence++;
78             map.put(key, occurrence);
79         } else {
80             map.put(key, 1);
81         }
82     }
83     //menghitung hasil peramlaan
84     Iterator iterator = map.keySet().iterator();
85     while (iterator.hasNext()) {
86         int key = (Integer) iterator.next();
87         int occurrence = map.get(key);
88         peramalan = (peramalan + (pusat_cluster1[(key-1)]*
89             occurrence)) ;
90     }
91     hasil_peramalan[i] = (peramalan/ countGroup[i]);
92 }
93 }

```

**Kode Program 5.10** Implementasi proses defuzzifikasi

### 5.3 Implementasi Antarmuka

Pada implementasi antarmuka dijelaskan beberapa halaman antarmuka dari program hasil implementasi. Implementasi antarmuka berdasarkan perancangan yang telah dijelaskan pada bab perancangan. Halaman antarmuka yang dibuat adalah halaman input, halaman tampil data, halaman proses, halaman hasil, dan halaman perhitungan *error*.

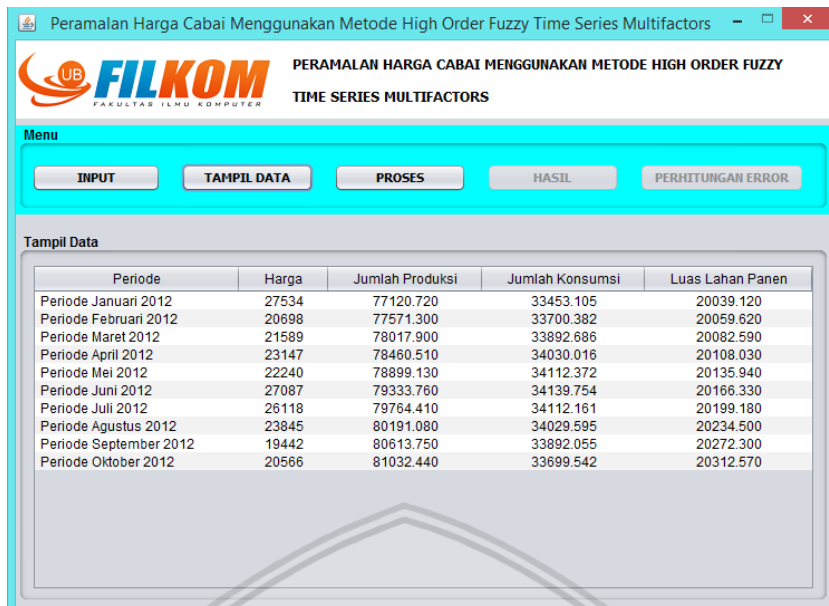
#### 5.3.1 Implementasi Antarmuka Halaman Input

Halaman input merupakan halaman yang pertama muncul setelah program dijalankan. Pada halaman input ini terdapat beberapa input yang harus di masukkan yaitu data input, data training, data *testing*, *order*, dan konstanta. Selain itu pada halaman input juga terdapat tiga *button* yaitu *button browse data*, *button fuzzification*, dan *button forecasting*. Tampilan antarmuka halaman input ditunjukkan pada Gambar 5.1.

Gambar 5.1 Implementasi antarmuka halaman input

#### 5.3.2 Implementasi Antarmuka Halaman Tampil Data

Halaman tampil data merupakan halaman yang tampil setelah melakukan input data. Halaman ini berisi tabel yang menampilkan data harga cabai, jumlah produksi, jumlah konsumsi, dan luas lahan panen. Tampilan antarmuka halaman tampil data ditunjukkan pada Gambar 5.2.

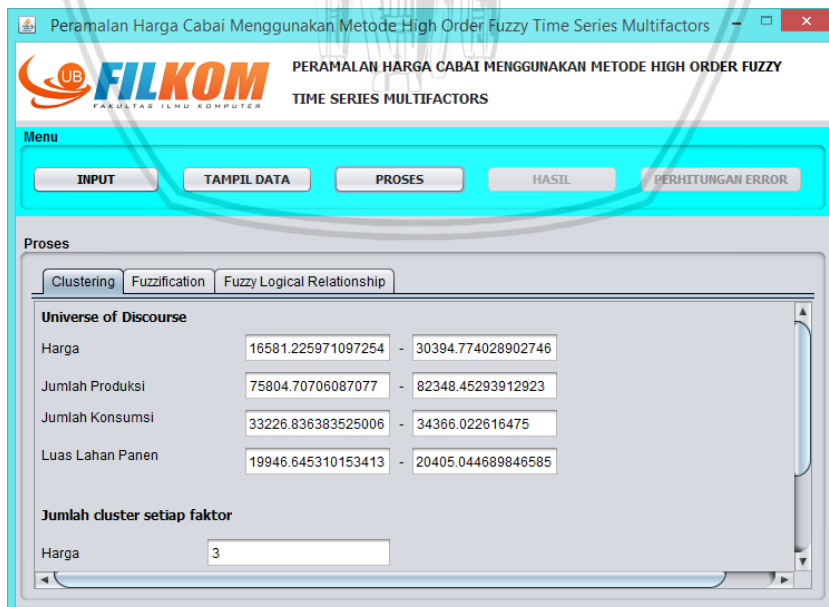


Periode	Harga	Jumlah Produksi	Jumlah Konsumsi	Luas Lahan Panen
Periode Januari 2012	27534	77120.720	33453.105	20039.120
Periode Februari 2012	20698	77571.300	33700.382	20059.620
Periode Maret 2012	21589	78017.900	33892.686	20082.590
Periode April 2012	23147	78460.510	34030.016	20108.030
Periode Mei 2012	22240	78899.130	34112.372	20135.940
Periode Juni 2012	27087	79333.760	34139.754	20166.330
Periode Juli 2012	26118	79764.410	34112.161	20199.180
Periode Agustus 2012	23845	80191.080	34029.595	20234.500
Periode September 2012	19442	80613.750	33892.055	20272.300
Periode Oktober 2012	20566	81032.440	33699.542	20312.570

Gambar 5.2 Implementasi antarmuka halaman tampil data

### 5.3.3 Implementasi Antarmuka Halaman Proses

Halaman proses merupakan halaman yang muncul setelah *button fuzzification* telah dieksekusi. Halaman proses ini terdiri dari tiga bagian atau tab yaitu *clustering*, *fuzzification*, dan *fuzzy logical relationship*. Pada halaman antarmuka *clustering* menampilkan *Universe of Discourse* dan jumlah *cluster* pada masing-masing faktor peramalan. Tampilan antarmuka halaman proses bagian *clustering* yang ditunjukkan pada Gambar 5.3.



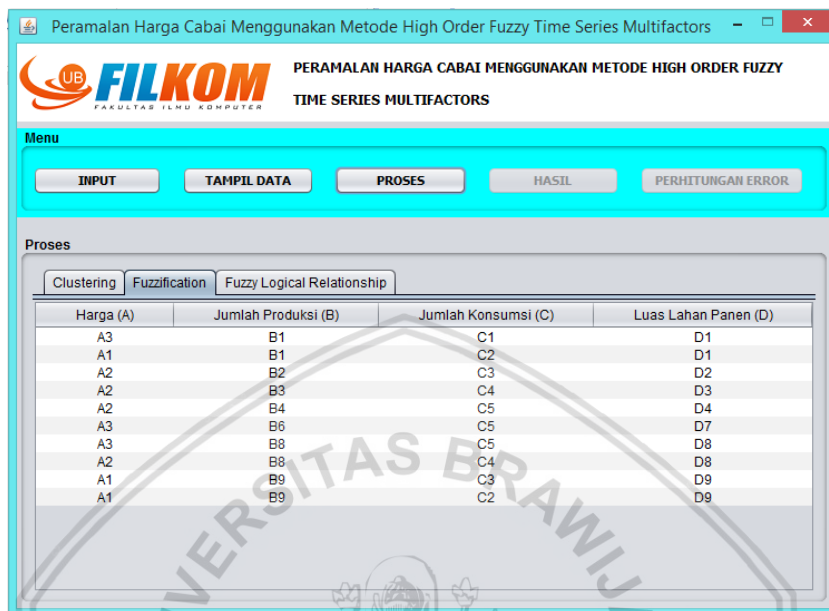
Universe of Discourse	
Harga	16581.225971097254 - 30394.774028902746
Jumlah Produksi	75804.70706087077 - 82348.45293912923
Jumlah Konsumsi	33226.836383525006 - 34366.022616475
Luas Lahan Panen	19946.645310153413 - 20405.044689846585

Jumlah cluster setiap faktor	
Harga	3

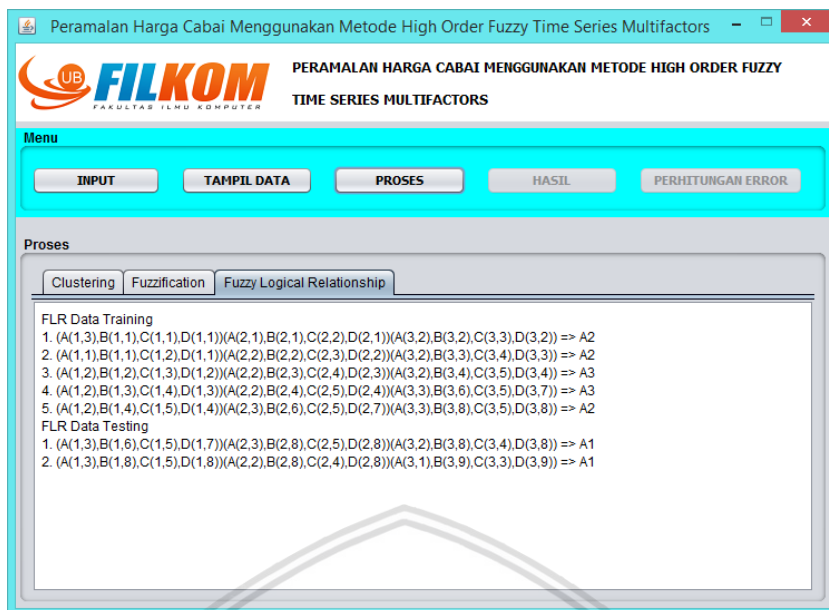
Gambar 5.3 Implementasi antarmuka halaman proses *clustering*

Pada tab selanjutnya yaitu pada tab *fuzzification*, tab ini berfungsi menampilkan hasil proses fuzzifikasi. Pada halaman ini terdapat tabel yang berisi hasil fuzzifikasi dari setiap faktor peramalan. Tampilan antarmuka halaman proses bagian fuzzifikasi yang ditunjukkan pada Gambar 5.4.



**Gambar 5.4** Implementasi antarmuka halaman proses fuzzifikasi

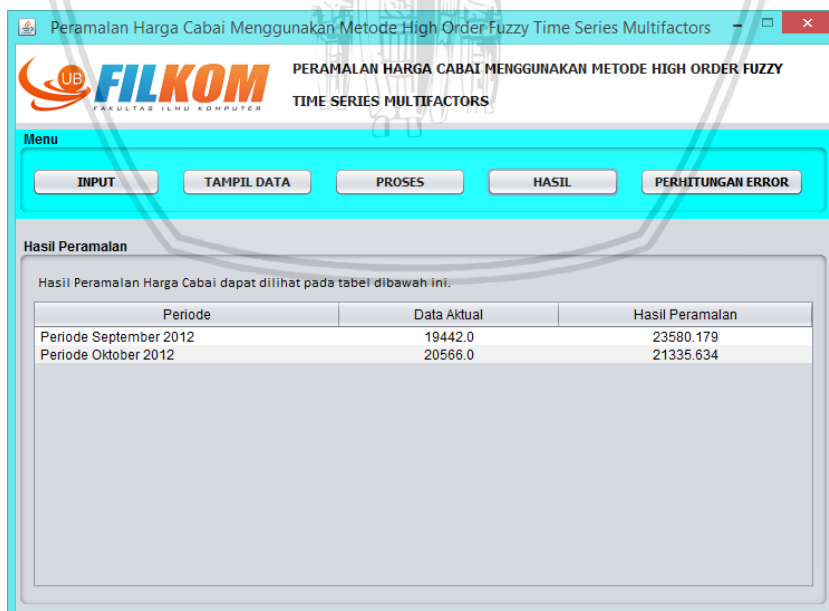
Pada tab *Fuzzy Logical Relationship* menampilkan hasil dari proses pembentukan *Fuzzy Logical Relationship*. Tampilan pada halaman ini menampilkan berupa *textarea* yang berisi dari *output* proses tersebut. Tampilan antarmuka halaman proses bagian *Fuzzy Logical Relationship* ditunjukkan pada Gambar 5.5.



**Gambar 5.5** Implementasi antarmuka halaman proses bagian *fuzzy logical relationship*

### 5.3.4 Implementasi Antarmuka Halaman Hasil

Pada implementasi antarmuka halaman hasil ini merupakan halaman yang menampilkan hasil dari proses defuzzifikasi. Halaman hasil ini menampilkan tabel yang berisi data aktual dan data dari hasil peramalan. Berikut merupakan tampilan antarmuka halaman hasil yang ditunjukkan pada Gambar 5.6.



**Gambar 5.6** Implementasi halaman antarmuka hasil

### 5.3.5 Implementasi Antarmuka Halaman Perhitungan *Error*

Pada implementasi antarmuka halaman perhitungan *error* merupakan halaman yang menampilkan hasil dari proses perhitungan *Mean Square Error* (MSE). Pada halaman antarmuka ini menampilkan hasil MSE pada *textfield*. Tampilan antarmuka halaman perhitungan *error* ditunjukkan pada Gambar 5.7.



Gambar 5.7 Implementasi halaman antarmuka perhitungan *error*



## BAB 6 PENGUJIAN DAN ANALISIS

bab ini membahas hasil pengujian dan analisis hasil pengujian dari hasil implementasi metode *High Order Fuzzy Times Series Multifactors* untuk peramalan harga cabai. Pada bab pengujian dan analisis ini disesuaikan dengan perancangan pengujian pada bab 4.

### 6.1 Sistematika Pengujian

Pada bab 4 mengenai perancangan telah dibahas perancangan pengujian, sudah disebutkan bahwa pengujian akan dilakukan sejumlah tiga tahap pengujian, yaitu pengujian pengaruh order, pengujian jumlah data *training*, dan pengujian validasi. Dari semua tahap pengujian tersebut akan dilakukan pengujian dan analisis bagaimana besar pengaruh terhadap hasil perhitungan *Mean Square Error* (MSE).

### 6.2 Hasil dan Analisis

Berikut hasil pengujian dan analisis yang disesuaikan dengan perancangan pengujian pada bab 4.

#### 6.2.1 Pengujian Pengaruh *Order* Terhadap MSE

Pada pengujian ini dilakukan pengujian untuk mengetahui pengaruh *order* terhadap nilai MSE. *Order* merupakan urutan jumlah dari data *time series* pada periode waktu  $t$  yang memengaruhi nilai data. Jadi jika nilai *order* adalah 3 maka peramalan akan melibatkan 3 data sebelumnya pada proses pembentukan *Fuzzy Logical Relationship* (FLR). Pada pengujian ini hanya nilai *order* yang nilainya diubah sedangkan variabel lain akan dibuat tetap. Pengujian akan dilakukan sebanyak 10 kali. Pada pengujian ini nilai variabel jumlah data *training* adalah 64 data yaitu periode Januari 2010 sampai April 2015, sedangkan jumlah data *testing* adalah 8 data yakni periode Mei 2015 sampai Desember 2015. Nilai konstanta yang digunakan adalah tetap yaitu 30. Berikut adalah hasil pengujian pengaruh *order* terhadap nilai MSE yang ditunjukkan pada Tabel 6.1.

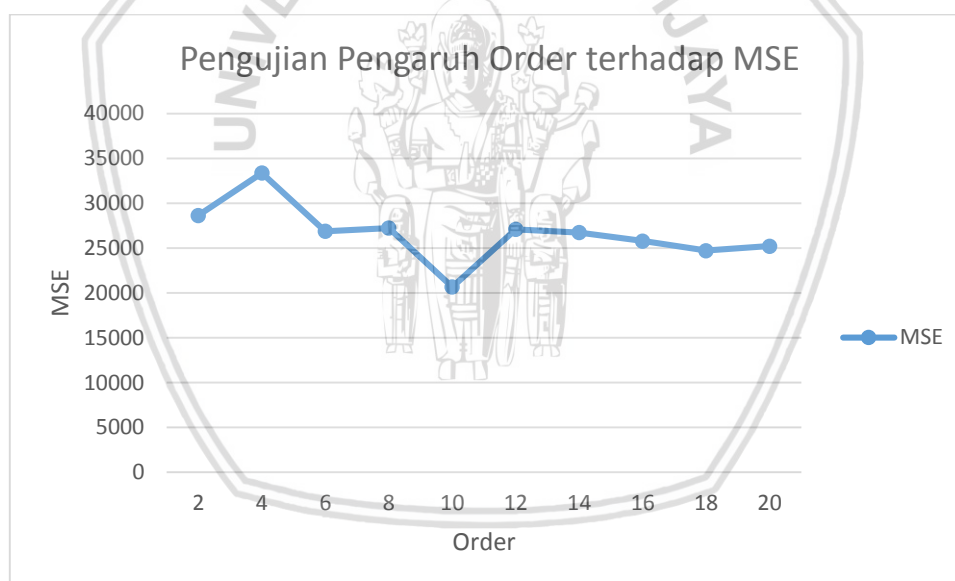
**Tabel 6.1** Hasil pengujian pengaruh *order* terhadap MSE

No	<i>Order</i>	MSE
1	2	28635,35203
2	4	33377,429472
3	6	26879,826306
4	8	27238,69987

5	10	20674,551427
6	12	27090,958723
7	14	26732,208059
8	16	25790,936794
9	18	24718,423558
10	20	25223,684739

Selengkapnya dapat dilihat pada Tabel di Lampiran 2.

Dari hasil pengujian *order* seperti yang telah ditunjukkan pada Tabel 6.1 maka pada percobaan pertama dengan *order* sebanyak 2 dan konstanta sebanyak 30 dapat diperoleh MSE sebesar 28635,35203. Selanjutnya pada percobaan kedua dengan *order* sebanyak 4 dan konstanta sebanyak 30 diperoleh MSE sebesar 33377,429472. Adapun perubahan nilai MSE dari setiap pengujian nilai *order* yang dilakukan digambarkan dalam grafik yang ditunjukkan pada Gambar 6.1.



**Gambar 6.1** Grafik hasil pengujian pengaruh *order* terhadap MSE

Berdasarkan Gambar 6.1 nilai *order* yang berbeda-beda akan memengaruhi besar nilai MSE. Dalam penelitian ini untuk menentukan *order* memiliki ketentuan yaitu nilai *order* harus lebih dari 0 dan batas maksimal order adalah setengah dari nilai data *training* (Lin, et al., 2009). Penelitian ini juga menunjukkan bahwa nilai *order* yang berbeda akan memengaruhi besarnya nilai MSE. Semakin besar nilai *order* tidak akan menjamin menghasilkan nilai MSE yang bagus, demikian pula jika semakin kecil nilai *order* tidak akan menjamin pula menghasilkan nilai MSE yang baik. Meskipun banyak penelitian yang menunjukkan hasil bahwa semakin besar

nilai *order* maka dapat membuat peramalan semakin akurat. Hal tersebut seperti ditunjukkan pada penelitian implementasi metode *Multifactors High order Fuzzy Times Series Model* untuk melakukan prediksi harga emas, yang menyatakan bahwa semakin besar nilai *order* maka semakin kecil tingkat kesalahannya (Wardhani, 2015).

### 6.2.2 Pengujian Pengaruh Data *Training* Terhadap MSE

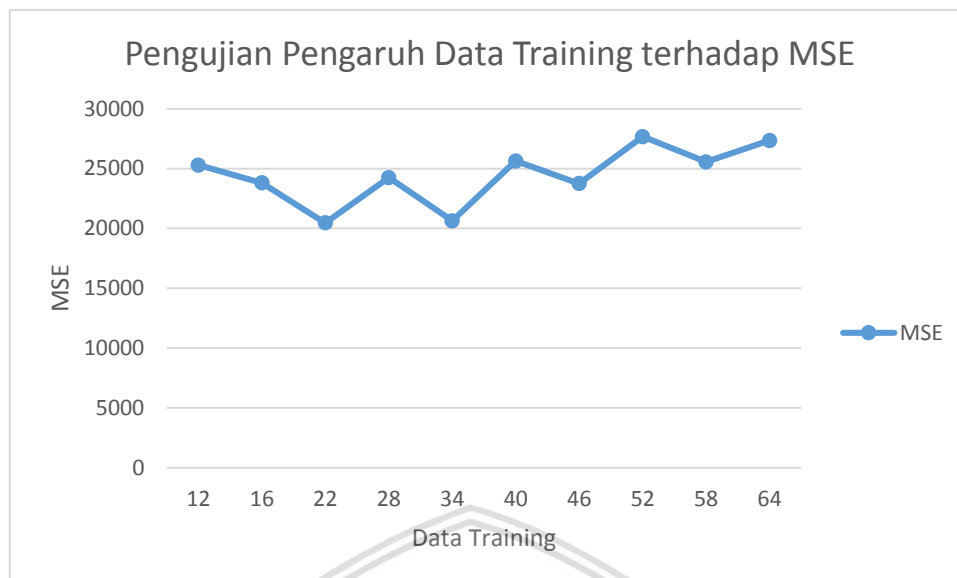
Pada pengujian ini akan dilakukan untuk mengetahui pengaruh jumlah data *training* terhadap nilai MSE. Pada pengujian ini jumlah data *training* yang digunakan berbeda-beda. Pengujian ini akan dilakukan sebanyak 10 kali. Pada pengujian ini jumlah data *testing* adalah 8, nilai *order* yang digunakan sebanyak 6. Berikut adalah hasil pengujian pengaruh jumlah data *training* terhadap MSE yang ditunjukkan pada Tabel 6.2.

**Tabel 6.2** Hasil pengujian pengaruh jumlah data *training* terhadap MSE

No	Jumlah Data <i>Training</i>	MSE
1	12	25297,846054
2	16	23812,246587
3	22	20456,503739
4	28	24247,814239
5	34	20639,498915
6	40	25640,74504
7	46	23757,242726
8	52	27675,553993
9	58	25556,520224
10	64	27351,67151

Selengkapnya dapat dilihat pada Tabel di Lampiran 3.

Dari hasil pengujian jumlah data *training* seperti yang telah ditunjukkan pada Tabel 6.3 maka pada percobaan pertama dengan data *training* sebanyak 12 dapat diperoleh MSE sebesar 25297,846054. Selanjutnya pada percobaan kedua dengan data *training* sebanyak 16 diperoleh MSE sebesar 23812,246587. Perubahan nilai MSE dari setiap percobaan digambarkan dalam grafik yang ditunjukkan pada Gambar 6.2.



**Gambar 6.2** Grafik hasil pengujian pengaruh data *training* terhadap MSE

Berdasarkan dari Gambar 6.2, dapat disimpulkan bahwa nilai MSE sangat bergantung pada jumlah data *training* yang digunakan. Semakin besar data *training* yang digunakan tidak akan menjamin nilai MSE yang diperoleh akan baik pula, begitu pula sebaliknya. Faktanya nilai MSE akan menjadi semakin kecil apabila menggunakan data *training* dalam jumlah besar dan data yang digunakan memiliki rentang nilai yang tidak jauh berbeda dengan data *testing* atau data yang akan diramalkan. Dari hasil percobaan diatas hasil MSE terbesar adalah 27675,553993 dan hasil MSE terkecil adalah 20456,503739.

### 6.2.3 Pengujian Validasi Sistem

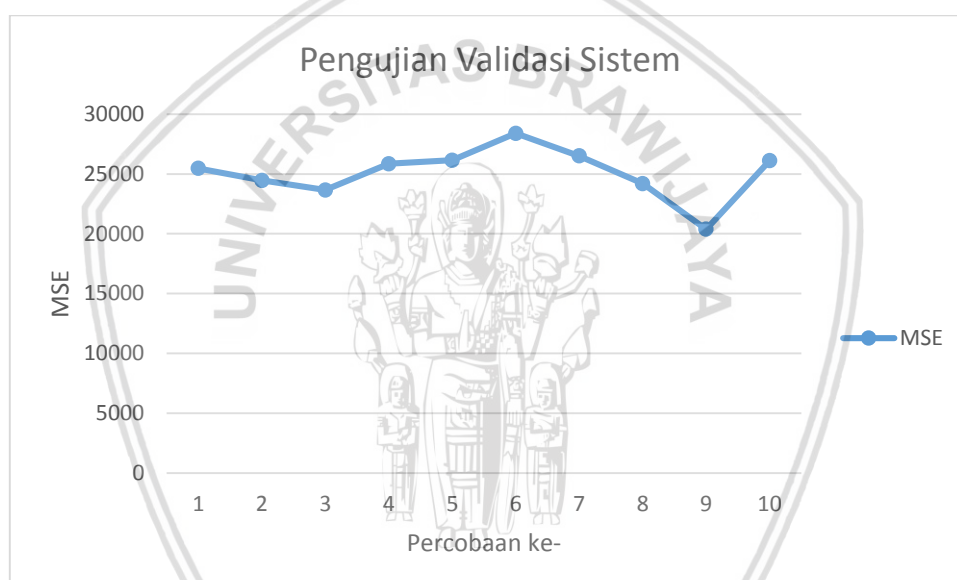
Pada pengujian ini dilakukan untuk mengetahui tingkat kestabilan dari sistem dengan menggunakan *order* dan data *training* terbaik dari pengujian sebelumnya. Pengujian ini akan dilakukan sebanyak sepuluh kali. Pada pengujian ini nilai data *training* menggunakan data yang sama agar dapat menunjukkan gambaran variasi dari nilai kesalahan MSE pada peramalan. Nilai *order* yang digunakan adalah 10, dan konstanta adalah 30. Berikut adalah hasil pengujian validasi sistem yang ditunjukkan pada Tabel 6.3.

**Tabel 6.3** Hasil pengujian validasi sistem

Percobaan ke-	MSE
1	25459,621345
2	24459,894575
3	23660,380525

4	25848,312905
5	26145,056755
6	28390,090879
7	26509,455174
8	24189,996599
9	20374,193709
10	26102,063593

Dari hasil pengujian validasi sistem seperti yang telah ditunjukkan pada Tabel 6.3 maka dapat digambarkan dengan grafik yang ditunjukkan oleh Gambar 6.3.



**Gambar 6.3** Grafik hasil pengujian validasi sistem

Berdasarkan dari grafik pada Gambar 6.3, dapat disimpulkan bahwa pengujian dari validasi sistem memiliki hasil MSE yang tidak terlalu jauh berbeda. Oleh karena itu, dapat disimpulkan bahwa implementasi dari metode *High Order Fuzzy Times Series Multifactors* yang digunakan sudah baik untuk digunakan sebagai metode peramalan.

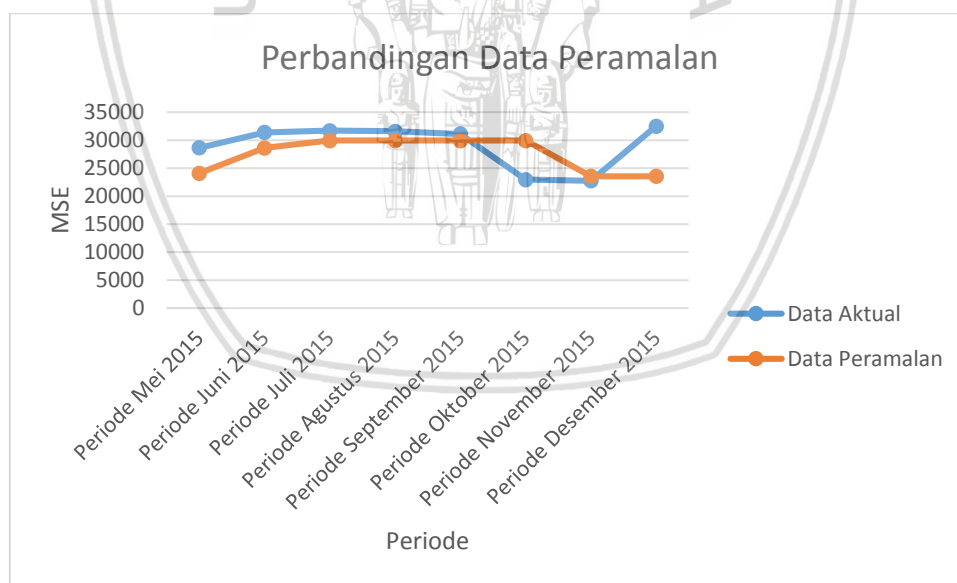
Berdasarkan dari serangkaian pengujian yang dilakukan diatas dapat diketahui bahwa nilai MSE terkecil terdapat pada pengujian dengan jumlah data *training* sebanyak 64 yaitu sebesar 20374,193709. Nilai *order* yang digunakan yaitu 10 dengan nilai konstanta adalah 30. Nilai tersebut diperoleh dari pengujian jumlah data *training* pada periode Januari 2010 sampai April 2015 untuk melakukan peramalan pada periode Mei 2015 sampai Desember 2015. Berikut

adalah perbandingan data aktual dengan hasil peramalan yang ditunjukkan pada Tabel 6.4.

**Tabel 6.4** Perbandingan data hasil peramalan

Periode	Data Aktual	Data Peramalan
Periode Mei 2015	28670,00	24097,908
Periode Juni 2015	31416,00	28630,746
Periode Juli 2015	31721,00	29936.209
Periode Agustus 2015	31596,00	29936.209
Periode September 2015	31104,00	29936.209
Periode Oktober 2015	22959,00	29936.209
Periode November 2015	22774,00	23569.199
Periode Desember 2015	32520,00	23569.199

Adapun grafik dari perbandingan data aktual dan hasil peramalan ditunjukkan pada Gambar 6.4.



**Gambar 6.4** Grafik perbandingan data aktual dengan hasil peramalan



## BAB 7 PENUTUP

Pada bab penutup terdiri dari dua sub bab yaitu kesimpulan dan saran. Kesimpulan adalah tahapan yang berisi hasil dari penelitian yang dilakukan disertai jawaban semua rumusan masalah yang sudah ditetapkan sebelumnya. Sedangkan saran adalah tahapan yang berisi hasil yang telah dicapai untuk memperbaiki kekurangan–kekurangan yang ada pada penelitian ini, serta memberikan pertimbangan pada pengembangan penelitian lebih lanjut.

### 7.1 Kesimpulan

Pada kesimpulan ini mengacu pada penelitian yang telah dilakukan yang meliputi perancangan, implementasi, serta pengujian dan analisis terhadap hasil implementasi peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* maka dapat diambil kesimpulan sebagai berikut.

1. Berdasarkan dari implementasi metode *High Order Fuzzy Times Series Multifactors* dalam melakukan peramalan harga cabai dilakukan dengan beberapa langkah yaitu langkah pertama perhitungan *Universe of Discourse* yang terdiri perhitungan *mean*, simpangan baku, dan perhitungan *Dmin* serta *Dmax*, langkah kedua yaitu perhitungan jumlah *cluster*, langkah ketiga yaitu perhitungan pusat *cluster*, langkah keempat yaitu pembentukan subinterval dan *fuzzy set*, langkah kelima yaitu proses fuzzifikasi, langkah keenam yaitu pembentukan *Fuzzy Logical Relationship* (FLR), langkah ketujuh adalah defuzzifikasi yang hasil *output*–nya merupakan hasil peramalan harga cabai.
2. Berdasarkan dari pengujian peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors*, maka dapat diketahui bahwa:
  - a. *Order* berpengaruh terhadap nilai *error*, semakin besar nilai *order* tidak akan menjamin menghasilkan nilai MSE yang bagus, demikian pula jika semakin kecil nilai *order* tidak akan menjamin pula menghasilkan nilai MSE yang bagus pula.
  - b. Data *training* berpengaruh terhadap nilai *error*, semakin besar data *training* yang digunakan tidak akan menjamin nilai MSE yang diperoleh akan baik, begitu pula sebaliknya. Faktanya nilai MSE ini cenderung akan menjadi semakin baik ketika memiliki tingkat kecocokan yang signifikan dengan data *testing* yang dipakai. Sehingga peramalan menggunakan metode *High Order Fuzzy Times Series Multifactors* memerlukan banyak data *training*.
  - c. Hasil implementasi peramalan harga cabai menggunakan metode *High Order Fuzzy Times Series Multifactors* baik digunakan untuk peramalan.

Dari pengujian yang telah dilakukan nilai MSE terkecil yang didapatkan sebesar 20374,19. Nilai *order* yang digunakan yaitu 10 dengan nilai konstanta



adalah 30. Nilai tersebut diperoleh dari pengujian jumlah data *training* pada periode Januari 2010 sampai April 2015.

## 7.2 Saran

Pada penelitian ini dilakukan peramalan harga cabai dengan menggunakan faktor yang memengaruhi hasil peramalan yaitu harga, jumlah produksi, jumlah konsumsi dan luas lahan panen. Diharapkan untuk penelitian selanjutnya yakni dengan melakukan penambahan faktor yang memengaruhi tingkat harga cabai. Peramalan pada objek lain juga diharapkan supaya mempertimbangkan faktor-faktor yang memengaruhinya. Pada penelitian selanjutnya juga dipertimbangkan untuk menambahkan data yang lebih banyak serta bervariasi sehingga nilai hasil perhitungan akurasi peramalan yang dihasilkan dalam penelitian dapat meningkat. Penggunaan data *training* yang banyak dan bervariasi dapat membuat pola data semakin terlihat, sehingga hasil peramalan yang dihasilkan memiliki akurasi yang lebih baik.



## Daftar Pustaka

- Badan Pusat Statistik. 2011. Badan Pusat Statistik. 2011, p. [www.bps.go.id/aboutus.php?pub=1&dse=1&pubs=21](http://www.bps.go.id/aboutus.php?pub=1&dse=1&pubs=21).
- Badan Pusat Statistik. 2015. Harga Cabai Nasional. 2015, pp. [http://www.bps.go.id/website/brs\\_ind/brsInd-20150803115537.pdf](http://www.bps.go.id/website/brs_ind/brsInd-20150803115537.pdf).
- Badan Pusat Statistik. 2016. *Sensus Penduduk 2010*. s.l. : Badan Pusat Statistik, 2016. pp. [http://www.bps.go.id/website/brs\\_ind/brsInd-20150803115537.pdf](http://www.bps.go.id/website/brs_ind/brsInd-20150803115537.pdf).
- Buffa S, et al. 1996. *Modern Production and Operation Management*. 8th Edition. London : John Willey and Sons Incorporated, 1996.
- Chen, Shyi Ming, Wang, N Y and Pan, J S. 2009. Forecasting Enrollments using Automatic Clustering Techniques and Fuzzy Logical Relationship. 2009, Vol. 36, pp. 11070-11076.
- Cheng, S H, Chen, S M and Jian, W S. 2016. Fuzzy Time Series Forecasting based on Fuzzy Logical Relationship and Similarity Measures. 2016, pp. 272-287.
- Efendi R, Ismail Z and Deris, M M. 2015. A New Linguistic Out-Sample Approach of Fuzzy Time Series for Daily Forecasting of Malaysian Electricity Load Demand. 2015, pp. 422-430.
- Farid, Miftah and Subekti, Nugroho Ari. 2012. Tinjauan Terhadap Produksi, Konsumsi, Distribusi dan Dinamika Harga Cabai di Indonesia. Desember 2012, Vol. 6 No. 2, pp. Hlm. 211-234.
- Fauziah, Normalita, Wahyuningsih, Sri and Nasution, Yuki Novia. 2016. Peramalan Menggunakan Fuzzy Time Series Chen (Studi Kasus: Curah Hujan Kota Samarinda). November 2016, Vol. 4 No. 2.
- Hsu, L Y. 2010. Temperature Prediction and TAIEX Forecasting Based on Fuzzy Relationship and MTPSO Techniques. 2010, pp. 2756-2770.
- Huang Y. 2011. A Hybrid Forecasting Model for Enrollments based on Aggregated Fuzzy Time Series and Particle Swarm Optimization. 2011, pp. 8014-8023.
- Imdadullah, Muhammad. 2013. Basic Statistic and Data Analysis. 2013, pp. <http://itfeature.com/time-series-analysis-andforecasting/time-series-analysis-forecasting>.

- Irsyad, Muhammad and Metarice, Sonya. 2013. Penerapan Metode Automatic Clustering dan High Order Fuzzy Time Series Pada Peramalan Curah Hujan. *Jurnal Sains dan Teknologi Industri*. 2013, Vol. 10 No. 2.
- Kusumadewi S and Purnomo H. 2013. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Yogyakarta : Graha Ilmu, 2013.
- Lin Yupei and Yang Y. 2009. Stock Markets Forecasting Based on Fuzzy Time Series Model. 2009, pp. 782-886.
- Makridakis S, et al. 1999. *Metode dan Aplikasi Peramalan Jilid I*. Edisi Kedua. Jakarta : Binarupa Aksara, 1999.
- Nugraha, Taufan. 2017. Peramalan Permintaan Daging Sapi Nasional Menggunakan Metode Multifactors High Order Fuzzy Time Series Model. 2017, Vol. 1 No. 12, pp. 1764-1770.
- Rofiq, Muhammad Aunu. 2017. *Peramalan Komoditas Strategis Pertanian Cabai Menggunakan Metode Backpropagation Neural Network*. Fakultas Teknologi Informasi : Institut Teknologi Sepuluh Nopember, 2017.
- Ross T. 2010. *Fuzzy Logic with Engineering Application*. 3th Edition. Chicester : John Willey and Sons, 2010.
- Saxeena P, K Sharma and S Easo. 2012. Forecasting Enrollments based on Fuzzy Time Series with Higher Forecast Accuracy Rate. 2012, pp. 957-961.
- Siagian, Dergibson and Sugiarto. 2002. *Metode Statistika untuk Bisnis dan Ekonomi*. Jakarta : Gramedia Pustaka Utama, 2002.
- Wang, N Y and Chen, Shyi Ming. 2009. Temperature Prediction and TAIEX Forecasting Based on Automatic Clustering Techniques and Two Factor High Order Fuzzy Time Series. 2009, pp. 2143-2154.
- Wardhani, Dessy Kusuma. 2015. *Implementasi Metode Multifactors High Order Fuzzy Time Series Model untuk Prediksi Harga Saham*. Malang : Fakultas Ilmu Komputer Universitas Brawijaya, 2015. Vol. 6 No. 6.